<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product overview</td>
<td>24</td>
</tr>
<tr>
<td>What's new</td>
<td>27</td>
</tr>
<tr>
<td>IBM WebSphere Portal new features and improvements</td>
<td>28</td>
</tr>
<tr>
<td>Portal Search - new key features and capabilities</td>
<td>31</td>
</tr>
<tr>
<td>Lotus Web Content Management new features and improvements</td>
<td>32</td>
</tr>
<tr>
<td>Documentation improvements and changes</td>
<td>34</td>
</tr>
<tr>
<td>Documentation resources</td>
<td>36</td>
</tr>
<tr>
<td>Versatile framework</td>
<td>39</td>
</tr>
<tr>
<td>Customization</td>
<td>40</td>
</tr>
<tr>
<td>Web content</td>
<td>44</td>
</tr>
<tr>
<td>Streamlined site creation</td>
<td>47</td>
</tr>
<tr>
<td>Portlets</td>
<td>50</td>
</tr>
<tr>
<td>Application integration</td>
<td>53</td>
</tr>
<tr>
<td>Collaboration</td>
<td>54</td>
</tr>
<tr>
<td>Accessibility features</td>
<td>56</td>
</tr>
<tr>
<td>Planning for WebSphere Portal</td>
<td>57</td>
</tr>
<tr>
<td>System requirements</td>
<td>59</td>
</tr>
<tr>
<td>WebSphere Portal Support Statement</td>
<td>60</td>
</tr>
<tr>
<td>User IDs and passwords</td>
<td>63</td>
</tr>
<tr>
<td>Server topologies</td>
<td>66</td>
</tr>
<tr>
<td>Single-server topology</td>
<td>68</td>
</tr>
<tr>
<td>Stand-alone server topology</td>
<td>69</td>
</tr>
<tr>
<td>Horizontal cluster topology</td>
<td>70</td>
</tr>
<tr>
<td>Vertical cluster topology</td>
<td>72</td>
</tr>
<tr>
<td>Combination of horizontal and vertical clusters</td>
<td>73</td>
</tr>
<tr>
<td>Multiple clusters</td>
<td>74</td>
</tr>
<tr>
<td>Single-server topology for Web Content Management</td>
<td>77</td>
</tr>
<tr>
<td>Dual-server configuration for Web Content Management</td>
<td>79</td>
</tr>
<tr>
<td>Staging-server topology for Web Content Management</td>
<td>81</td>
</tr>
<tr>
<td>Web Content Management environments</td>
<td>83</td>
</tr>
<tr>
<td>Web content system overview</td>
<td>84</td>
</tr>
<tr>
<td>Web content authoring environments</td>
<td>86</td>
</tr>
<tr>
<td>Web content staging environments</td>
<td>89</td>
</tr>
<tr>
<td>Built-in redundancy</td>
<td>90</td>
</tr>
<tr>
<td>Web content testing environments</td>
<td>92</td>
</tr>
<tr>
<td>Web content delivery environments</td>
<td>94</td>
</tr>
<tr>
<td>Web servers</td>
<td>96</td>
</tr>
<tr>
<td>Database considerations</td>
<td>98</td>
</tr>
<tr>
<td>Database topologies</td>
<td>100</td>
</tr>
<tr>
<td>Shared database domains</td>
<td>102</td>
</tr>
<tr>
<td>Database compatibility considerations</td>
<td>105</td>
</tr>
<tr>
<td>Planning for DB2 on Windows</td>
<td>106</td>
</tr>
<tr>
<td>Planning for DB2 on UNIX or Linux</td>
<td>109</td>
</tr>
<tr>
<td>Planning for DB2 for i5/OS</td>
<td>112</td>
</tr>
<tr>
<td>Planning for DB2 for z/OS</td>
<td>115</td>
</tr>
<tr>
<td>Planning for Oracle</td>
<td>119</td>
</tr>
<tr>
<td>Topic</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Planning for Oracle RAC</td>
<td>121</td>
</tr>
<tr>
<td>Planning for SQL Server 2005</td>
<td>123</td>
</tr>
<tr>
<td>User registry considerations</td>
<td>125</td>
</tr>
<tr>
<td>Overview of user registry options</td>
<td>127</td>
</tr>
<tr>
<td>Virtual Member Manager integration</td>
<td>131</td>
</tr>
<tr>
<td>Realm support</td>
<td>133</td>
</tr>
<tr>
<td>Property extension</td>
<td>134</td>
</tr>
<tr>
<td>Security and authentication considerations</td>
<td>135</td>
</tr>
<tr>
<td>Authentication</td>
<td>137</td>
</tr>
<tr>
<td>Federal Information Processing Standards</td>
<td>138</td>
</tr>
<tr>
<td>Planning for single sign-on</td>
<td>140</td>
</tr>
<tr>
<td>Secure Socket Layer</td>
<td>141</td>
</tr>
<tr>
<td>Credential Vault</td>
<td>142</td>
</tr>
<tr>
<td>Caching considerations</td>
<td>144</td>
</tr>
<tr>
<td>Planning for external security managers</td>
<td>145</td>
</tr>
<tr>
<td>Cluster considerations</td>
<td>149</td>
</tr>
<tr>
<td>Guidelines for setting up a cluster</td>
<td>151</td>
</tr>
<tr>
<td>Limitations for setting up a cluster</td>
<td>153</td>
</tr>
<tr>
<td>HTTP session failover</td>
<td>154</td>
</tr>
<tr>
<td>Setting up an i5/OS database in a cluster</td>
<td>156</td>
</tr>
<tr>
<td>Security options</td>
<td>159</td>
</tr>
<tr>
<td>Security Scenarios</td>
<td>161</td>
</tr>
<tr>
<td>Using external security managers in a cluster</td>
<td>163</td>
</tr>
<tr>
<td>Planning for multiple clusters</td>
<td>165</td>
</tr>
<tr>
<td>Database sharing between multiple clusters</td>
<td>168</td>
</tr>
<tr>
<td>WebSphere Virtual Enterprise Dynamic Clusters</td>
<td>170</td>
</tr>
<tr>
<td>Cluster maintenance</td>
<td>172</td>
</tr>
<tr>
<td>WebSphere Process Server integration</td>
<td>174</td>
</tr>
<tr>
<td>Prerequisites for setting up process portals</td>
<td>175</td>
</tr>
<tr>
<td>Understanding configuration options for process integration</td>
<td>177</td>
</tr>
<tr>
<td>Planning to use process portals</td>
<td>180</td>
</tr>
<tr>
<td>Release notes</td>
<td>181</td>
</tr>
<tr>
<td>Installing WebSphere Portal</td>
<td>182</td>
</tr>
<tr>
<td>Installation methods, options, and sources</td>
<td>184</td>
</tr>
<tr>
<td>Installation options</td>
<td>185</td>
</tr>
<tr>
<td>Installation methods</td>
<td>186</td>
</tr>
<tr>
<td>Installation source location</td>
<td>189</td>
</tr>
<tr>
<td>Setting up a single server</td>
<td>192</td>
</tr>
<tr>
<td>Preparing your Windows operating system</td>
<td>194</td>
</tr>
<tr>
<td>Installing WebSphere Portal on Windows</td>
<td>195</td>
</tr>
<tr>
<td>Changing to developer mode</td>
<td>199</td>
</tr>
<tr>
<td>Configuring developer mode on Windows</td>
<td>201</td>
</tr>
<tr>
<td>Installing DB2 on the same server as WebSphere Portal</td>
<td>203</td>
</tr>
<tr>
<td>Modifying database properties</td>
<td>204</td>
</tr>
<tr>
<td>Setting up databases</td>
<td>207</td>
</tr>
<tr>
<td>Configuring JCR collation support</td>
<td>209</td>
</tr>
<tr>
<td>Assigning custom table spaces</td>
<td>211</td>
</tr>
</tbody>
</table>
Configuring databases ................................................................. 407
Preparing DB2 on AIX for a stand-alone production server .......... 409
Installing DB2 ........................................................................ 411
Create users ........................................................................... 412
Creating remote databases ......................................................... 414
Assigning custom table spaces ................................................... 417
Modifying database properties .................................................. 419
Configuring JCR collation support .............................................. 422
Configure WebSphere Portal to use DB2 ...................................... 424
Configuring DB2 for large file handling in Web Content Management ........................................................................ 427
Changing driver types ................................................................. 428
Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server ............................................................ 430
Installing DB2 ........................................................................... 432
Create users ........................................................................... 433
Creating remote databases ......................................................... 435
Assigning custom table spaces ................................................... 438
Modifying database properties .................................................. 440
Configuring JCR collation support .............................................. 443
Configure WebSphere Portal to use DB2 ...................................... 445
Configuring DB2 for large file handling in Web Content Management ........................................................................ 448
Changing driver types ................................................................. 449
Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server ........................................................................ 451
Installing Oracle ........................................................................ 452
Creating databases ..................................................................... 453
Creating users .......................................................................... 455
Modifying database properties .................................................. 457
Setting up databases ................................................................... 460
HP-UX stand-alone: Creating JCR table spaces ................................ 461
Assigning custom table spaces ................................................... 463
Configure WebSphere Portal to use Oracle ................................. 465
Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server ........................................................................ 467
Installing Oracle RAC ................................................................. 468
Create users ........................................................................... 469
Creating databases ..................................................................... 471
Modifying database properties .................................................. 473
Setting up databases ................................................................... 476
HP-UX stand-alone: Creating JCR table spaces ................................ 477
Assigning custom table spaces ................................................... 479
Configuring WebSphere Portal to use Oracle RAC .......................... 481
Setting up a remote SQL Server database on Windows for a stand-alone production server ............................................................ 484
Installing SQL Server ................................................................. 485
Modifying database properties .................................................. 488
Setting up databases ................................................................... 491
Creating users manually .............................................................. 492
HP-UX stand-alone server: Assigning custom filegroups on SQL Server 2005 ............................................................ 494
Configuring WebSphere Portal to use SQL Server 2005 .................................................. 496
Verifying databases ................................................................................................. 498
Preparing a remote Web server when portal is installed on HP-UX ................... 499
Configuring WebSphere Portal to use a user registry on HP-UX ......................... 501
Preparing user registries on HP-UX ................................................................... 503
Preparing a Domino Directory server ................................................................. 504
Preparing a SecureWay Security Server ............................................................ 506
Preparing a Novell eDirectory ............................................................................ 507
Preparing a Sun Java System Directory Server .................................................. 508
Preparing a Tivoli Directory Server .................................................................. 509
Choosing your user registry model on HP-UX .................................................... 510
Configuring a stand-alone LDAP user registry on HP-UX ................................. 511
Configuring a stand-alone LDAP user registry on HP-UX .................................. 512
Configuring a stand-alone LDAP user registry over SSL on HP-UX .................. 515
Configuring the default federated repository on HP-UX ...................................... 519
Configuring a federated LDAP user registry on HP-UX ....................................... 520
Adding an LDAP user registry on HP-UX ............................................................ 521
Adding an LDAP user registry over SSL on HP-UX ............................................ 526
Adding a database user registry on HP-UX ......................................................... 532
Adding realm support on HP-UX ...................................................................... 536
Adapting the attribute configuration .................................................................. 539
Querying the defined attributes on HP-UX ......................................................... 541
Adding attributes on HP-UX ............................................................................. 542
Mapping attributes ............................................................................................. 544
Removing attributes ........................................................................................... 548
Stand-alone server: Configuring WebSphere Portal to use dynamic groups on HP-UX ................................................................. 550
Stand-alone server: Enabling referrals for your LDAP user registry on HP-UX ........ 552
Tune your servers ................................................................................................. 553
Setting up a stand-alone server on i5/OS ............................................................... 554
Preparing your IBM i5/OS ............................................................................... 555
Installing WebSphere Portal on IBM i5/OS .......................................................... 556
Preparing DB2 for i5/OS for a stand-alone production server ............................. 560
Preparing for creation of databases .................................................................... 561
Create user profiles ............................................................................................. 564
Creating remote databases .................................................................................. 565
Modifying database properties .......................................................................... 567
Transferring DB2 for i5/OS manually ................................................................. 570
Verifying database connections ......................................................................... 572
Preparing the Web server when portal is installed on IBM i5/OS ....................... 574
Configuring WebSphere Portal to use a user registry on IBM i5/OS .................... 576
Preparing a Tivoli Directory Server for IBM i5/OS ............................................ 578
Choosing your user registry model on IBM i5/OS ............................................... 579
Configuring a stand-alone LDAP user registry on IBM i5/OS ......................... 580
Configuring a stand-alone LDAP user registry on IBM i5/OS ............................ 581
Configuring a stand-alone LDAP user registry over SSL on IBM i5/OS ............. 584
Configuring the default federated repository on IBM i5/OS ............................... 588
Configuring a federated LDAP user registry on IBM i5/OS ............................... 589
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adding an LDAP user registry on IBM i5/OS</td>
<td>590</td>
</tr>
<tr>
<td>Adding an LDAP user registry over SSL on IBM i5/OS</td>
<td>595</td>
</tr>
<tr>
<td>Adding a database user registry on IBM i5/OS</td>
<td>601</td>
</tr>
<tr>
<td>Adding realm support on IBM i5/OS</td>
<td>604</td>
</tr>
<tr>
<td>Adapting the attribute configuration</td>
<td>607</td>
</tr>
<tr>
<td>Querying the defined attributes on IBM i5/OS</td>
<td>609</td>
</tr>
<tr>
<td>Adding attributes on IBM i5/OS</td>
<td>610</td>
</tr>
<tr>
<td>Mapping attributes on IBM i5/OS</td>
<td>612</td>
</tr>
<tr>
<td>Removing attributes</td>
<td>616</td>
</tr>
<tr>
<td>Stand-alone server: Configuring WebSphere Portal to use dynamic groups on i5/OS</td>
<td>618</td>
</tr>
<tr>
<td>Stand-alone server: Enabling referrals for your LDAP user registry on i5/OS</td>
<td>620</td>
</tr>
<tr>
<td>Tune your servers</td>
<td>621</td>
</tr>
<tr>
<td>Setting up a stand-alone server on Linux</td>
<td>622</td>
</tr>
<tr>
<td>Preparing your Linux operating system</td>
<td>623</td>
</tr>
<tr>
<td>Installing WebSphere Portal on Linux</td>
<td>624</td>
</tr>
<tr>
<td>Configuring databases</td>
<td>628</td>
</tr>
<tr>
<td>Preparing DB2 on AIX for a stand-alone production server</td>
<td>630</td>
</tr>
<tr>
<td>Installing DB2</td>
<td>632</td>
</tr>
<tr>
<td>Create users</td>
<td>633</td>
</tr>
<tr>
<td>Creating remote databases</td>
<td>635</td>
</tr>
<tr>
<td>Assigning custom table spaces</td>
<td>638</td>
</tr>
<tr>
<td>Modifying database properties</td>
<td>640</td>
</tr>
<tr>
<td>Configuring JCR collation support</td>
<td>643</td>
</tr>
<tr>
<td>Configure WebSphere Portal to use DB2</td>
<td>645</td>
</tr>
<tr>
<td>Configuring DB2 for large file handling in Web Content Management</td>
<td>648</td>
</tr>
<tr>
<td>Changing driver types</td>
<td>649</td>
</tr>
<tr>
<td>Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server</td>
<td>651</td>
</tr>
<tr>
<td>Installing DB2</td>
<td>653</td>
</tr>
<tr>
<td>Create users</td>
<td>654</td>
</tr>
<tr>
<td>Creating remote databases</td>
<td>656</td>
</tr>
<tr>
<td>Assigning custom table spaces</td>
<td>659</td>
</tr>
<tr>
<td>Modifying database properties</td>
<td>661</td>
</tr>
<tr>
<td>Configuring JCR collation support</td>
<td>664</td>
</tr>
<tr>
<td>Configure WebSphere Portal to use DB2</td>
<td>666</td>
</tr>
<tr>
<td>Configuring DB2 for large file handling in Web Content Management</td>
<td>669</td>
</tr>
<tr>
<td>Changing driver types</td>
<td>670</td>
</tr>
<tr>
<td>Preparing DB2 for z/OS for a stand-alone production server</td>
<td>672</td>
</tr>
<tr>
<td>Installing DB2 for z/OS</td>
<td>673</td>
</tr>
<tr>
<td>Creating users</td>
<td>674</td>
</tr>
<tr>
<td>Creating remote databases</td>
<td>677</td>
</tr>
<tr>
<td>Creating the Java Content Repository database</td>
<td>679</td>
</tr>
<tr>
<td>Assigning custom table spaces</td>
<td>685</td>
</tr>
<tr>
<td>Modifying database properties</td>
<td>687</td>
</tr>
<tr>
<td>Configuring WebSphere Portal to use DB2 for z/OS</td>
<td>691</td>
</tr>
<tr>
<td>Verifying database connections</td>
<td>694</td>
</tr>
<tr>
<td>Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server</td>
<td>696</td>
</tr>
</tbody>
</table>
Verifying databases ................................................................. 977
Setting up a remote DB2 database ............................................. 978
Installing DB2 ......................................................................... 980
Create users ........................................................................... 981
Creating remote databases ....................................................... 983
Assigning custom table spaces ................................................ 987
Modifying database properties .................................................. 989
Configuring JCR collation support ............................................. 992
Configuring WebSphere Portal to use DB2 ............................... 994
Configuring DB2 for large file handling in Web Content Management .................................................. 997
Changing driver types ............................................................. 998
Setting up a remote Oracle database on Windows for a stand-alone production server .............................. 1000
Installing Oracle ..................................................................... 1001
Create users ........................................................................... 1002
Creating databases ................................................................... 1004
Modifying database properties ................................................ 1006
Setting up databases ............................................................... 1009
Windows stand-alone server: Creating JCR table spaces ............. 1010
Assigning custom table spaces ................................................ 1012
Configuring WebSphere Portal to use Oracle ......................... 1014
Setting up a remote Oracle RAC database on Windows for a stand alone production server ..................... 1016
Installing Oracle RAC ............................................................. 1017
Create users ........................................................................... 1018
Creating databases ................................................................... 1020
Modifying database properties ................................................ 1021
Setting up databases ............................................................... 1024
Windows stand-alone server: Creating JCR table spaces ............. 1025
Assigning custom table spaces ................................................ 1027
Configure WebSphere Portal to use Oracle RAC ....................... 1029
Setting up a remote SQL Server database on Windows for a stand-alone production server ...................... 1032
Installing SQL Server .............................................................. 1033
Modifying database properties ................................................ 1036
Setting up databases ............................................................... 1039
Creating users manually .......................................................... 1040
Windows stand-alone server: Assigning custom filegroups on SQL Server 2005 ........................................... 1042
Configuring WebSphere Portal to use SQL Server 2005 ......... 1044
Verifying databases ............................................................... 1046
Preparing a remote Web server when portal is installed on Windows ......................................................... 1047
Configuring WebSphere Portal to use a user registry on Windows ............................................................. 1049
Preparing user registries on Windows ....................................... 1051
Preparing an Active Directory server ........................................ 1052
Preparing an Active Directory-Lightweight-Directory-Services on Windows .................................................. 1054
Preparing a Domino Directory server ....................................... 1055
Preparing a SecureWay Security Server .................................... 1057
Preparing a Novell eDirectory .................................................. 1058
Preparing a Sun Java System Directory Server ......................... 1059
Preparing a Tivoli Directory Server ............................................ 1060
Modifying database properties ................................................................. 1359
Setting up databases .................................................................................. 1362
Creating JCR table spaces ......................................................................... 1363
Assigning custom table spaces .................................................................. 1365
Configure WebSphere Portal to use Oracle .................................................. 1367
Configuring WebSphere Portal to use Oracle RAC ........................................ 1369
Installing Oracle RAC .................................................................................. 1370
Create users ................................................................................................. 1371
Creating databases ...................................................................................... 1373
Modifying database properties .................................................................... 1375
Setting up databases .................................................................................... 1378
Creating JCR table spaces ........................................................................... 1379
Assigning custom table spaces .................................................................... 1381
Configuring WebSphere Portal to use Oracle RAC ........................................ 1383
Configuring WebSphere Portal to use SQL Server ....................................... 1386
Installing SQL Server .................................................................................. 1387
Modifying database properties .................................................................... 1390
Setting up databases .................................................................................... 1393
Assigning custom filegroups ........................................................................ 1394
Configuring WebSphere Portal to use SQL Server 2005 .............................. 1396
Verifying databases ...................................................................................... 1398
Configuring the primary node to communicate with the deployment manager on HP-UX .......................................................... 1399
Removing search collections on HP-UX ...................................................... 1400
Choosing the type of cluster to create on HP-UX ........................................ 1401
Creating a static cluster on HP-UX .............................................................. 1402
Creating a new dynamic cluster on HP-UX using WebSphere Virtual Enterprise .................................................................................. 1405
Preparing a remote Web server when portal is installed on HP-UX ............... 1409
Configuring WebSphere Portal to use a user registry on HP-UX in a clustered environment .......................................................... 1411
Preparing user registries on HP-UX ............................................................. 1412
Preparing a Domino Directory server .......................................................... 1413
Preparing a SecureWay Security Server ......................................................... 1415
Preparing a Novell eDirectory ..................................................................... 1416
Preparing a Sun Java System Directory Server ............................................. 1417
Preparing a Tivoli Directory Server ............................................................... 1418
Choosing your user registry model on HP-UX in a clustered environment .... 1419
Configuring a stand-alone LDAP user registry on HP-UX in a clustered environment .......................................................... 1420
Configuring a stand-alone LDAP user registry on HP-UX in a clustered environment .......................................................... 1421
Configuring a stand-alone LDAP user registry over SSL on HP-UX in a clustered environment .......................................................... 1424
Configuring the default federated repository on HP-UX in a clustered environment .......................................................... 1428
Configuring a federated LDAP user registry on HP-UX in a clustered environment .......................................................... 1429
Adding an LDAP user registry on HP-UX in a clustered environment ............ 1430
Adding an LDAP user registry over SSL on HP-UX in a clustered environment .......................................................... 1435
Adding a database user registry on HP-UX in a clustered environment ............ 1441
Adding realm support on HP-UX in a clustered environment ......................... 1445
Adapting the attribute configuration .............................................................. 1448
Querying the defined attributes on HP-UX .................................................. 1450
Adding attributes on HP-UX in a clustered environment ...................................................... 1451
Mapping attributes on HP-UX in a clustered environment ..................................................... 1453
Removing attributes ............................................................................................................. 1457
Preparing additional nodes on HP-UX ............................................................................... 1459
Installing WebSphere Portal on HP-UX on the additional nodes ............................................ 1460
Choosing the type of additional node to create on HP-UX ...................................................... 1464
Adding additional nodes to the static cluster on HP-UX ......................................................... 1465
Adding an additional node to an existing dynamic cluster ..................................................... 1468
Choosing the type of vertical cluster to create on HP-UX ...................................................... 1472
Adding vertical cluster members to a static cluster on HP-UX .............................................. 1473
Adding vertical cluster members to a dynamic cluster on HP-UX ........................................... 1475
Tune your servers .................................................................................................................. 1477
Configuring search in a cluster on HP-UX ............................................................................ 1478
Configuring Portal Search in a cluster on HP-UX ................................................................. 1479
Configuring JCR search in a cluster on HP-UX .................................................................... 1480
Setting up multiple clusters on HP-UX ............................................................................... 1482
Installing multiple clusters in a single cell on HP-UX ............................................................ 1483
Routing requests across clusters on HP-UX ........................................................................ 1489
Sharing database domains between clusters on HP-UX ......................................................... 1490
Setting up a cluster on IBM i5/OS ......................................................................................... 1492
Preparing prerequisite and corequisite software on IBM i5/OS ............................................ 1494
Preparing the WebSphere Application Server Deployment Manager on i5/OS .................... 1495
Preparing the Web server when portal is installed on IBM i5/OS ........................................... 1497
Preparing your IBM i5/OS .................................................................................................... 1499
Preparing the primary node on IBM i5/OS ........................................................................... 1500
Installing WebSphere Portal on IBM i5/OS on the primary node ....................................... 1501
Configure WebSphere Portal to use a remote database ......................................................... 1505
Preparing DB2 for i5/OS ....................................................................................................... 1506
Preparing for creation of databases ..................................................................................... 1507
Create user profiles ............................................................................................................. 1510
Creating remote databases ................................................................................................. 1511
Modifying database properties ......................................................................................... 1513
Transferring DB2 for i5/OS manually ................................................................................... 1516
Verifying database connections .......................................................................................... 1518
Configuring the primary node to communicate with the deployment manager on i5/OS ...... 1520
Removing search collections on IBM i5/OS ....................................................................... 1521
Creating the cluster on i5/OS ............................................................................................. 1522
Preparing the Web server when portal is installed on IBM i5/OS ........................................ 1526
Preparing user registries on IBM i5/OS .............................................................................. 1528
Preparing a Tivoli Directory Server for IBM i5/OS ............................................................ 1529
Configuring WebSphere Portal to use a user registry on IBM i5/OS in a clustered environment ................................................................. 1530
Choosing your user registry model on IBM i5/OS in a clustered environment ..................... 1531
Configuring a stand-alone LDAP user registry on IBM i5/OS in a clustered environment .... 1532
Configuring a stand-alone LDAP user registry on IBM i5/OS in a clustered environment .... 1533
Configuring a stand-alone LDAP user registry over SSL on IBM i5/OS in a clustered environment ................................................................. 1536
Configuring the default federated repository on IBM i5/OS in a clustered environment .......................... 1540
Configuring a federated LDAP user registry on IBM i5/OS in a clustered environment ....................... 1541
Adding an LDAP user registry on IBM i5/OS in a clustered environment ........................................... 1542
Adding an LDAP user registry over SSL on IBM i5/OS in a clustered environment ............................. 1547
Adding a database user registry on IBM i5/OS in a clustered environment ....................................... 1553
Adding realm support on IBM i5/OS in a clustered environment ....................................................... 1556
Adapting the attribute configuration ........................................... 1559
Querying the defined attributes on IBM i5/OS ...................................................................................... 1561
Adding attributes on IBM i5/OS in a clustered environment ................................................................. 1562
Mapping attributes on IBM i5/OS in a clustered environment ............................................................... 1564
Removing attributes ............................................................................................................................... 1568
Preparing additional nodes on IBM i5/OS ............................................................................................... 1570
Installing WebSphere Portal on i5/OS on the additional nodes ............................................................ 1571
Adding additional nodes to the cluster on i5/OS ..................................................................................... 1573
Adding vertical cluster members to a static cluster on IBM i5/OS .......................................................... 1576
Rendering documents on IBM i5/OS ....................................................................................................... 1578
Configuring an HTML rendering server on IBM i5/OS ........................................................................ 1579
Associating an HTML rendering server with WebSphere Portal .......................................................... 1580
Tune your servers ..................................................................................................................................... 1582
Configuring search in a cluster on i5/OS ................................................................................................. 1583
Configuring Portal Search in a cluster on i5/OS ...................................................................................... 1584
Configuring JCR search in a cluster on i5/OS ......................................................................................... 1585
Setting up multiple clusters on IBM i5/OS .............................................................................................. 1587
Installing multiple clusters in a single cell on IBM i5/OS ..................................................................... 1588
Routing requests across clusters on IBM i5/OS ..................................................................................... 1591
Sharing database domains between clusters on IBM i ......................................................................... 1592
Setting up a cluster on Linux .................................................................................................................. 1594
Preparing prerequisite and corequisite software on Linux ................................................................ 1596
Preparing the WebSphere Application Server Deployment Manager on Linux .......................... 1597
Preparing a remote Web server when portal is installed on Linux ...................................................... 1599
Preparing your Linux operating system ............................................................................................... 1601
Preparing the primary node on Linux .................................................................................................... 1602
Installing WebSphere Portal on Linux on the primary node ............................................................. 1603
Configure WebSphere Portal to use a remote database ................................................................. 1607
Configuring WebSphere Portal to use DB2 ......................................................................................... 1609
Installing DB2 ....................................................................................................................................... 1610
Create users ............................................................................................................................................ 1612
Creating remote databases ..................................................................................................................... 1614
Assigning custom table spaces ............................................................................................................... 1617
Modifying database properties ............................................................................................................. 1619
Configuring JCR collation support ......................................................................................................... 1622
Configure WebSphere Portal to use DB2 ................................................................................................. 1624
Configuring DB2 for large file handling in Web Content Management ........................................... 1627
Changing driver types ............................................................................................................................ 1628
Configuring WebSphere Portal to use DB2 for z/OS .......................................................................... 1630
Installing DB2 for z/OS ......................................................................................................................... 1631
Creating users ......................................................................................................................................... 1632
Creating users ........................................................................................................... 2023
Creating remote databases ..................................................................................... 2026
Creating the Java Content Repository database .................................................. 2028
Assigning custom table spaces .............................................................................. 2034
Configuring WebSphere Portal to use DB2 for z/OS ........................................... 2036
Configuring WebSphere Portal to use Oracle ....................................................... 2039
Installing Oracle .................................................................................................... 2040
Creating users ........................................................................................................ 2041
Creating databases ................................................................................................ 2043
Modifying database properties .............................................................................. 2045
Setting up databases .............................................................................................. 2048
Creating JCR table spaces ..................................................................................... 2049
Assigning custom table spaces ............................................................................ 2051
Configuring WebSphere Portal to use Oracle ....................................................... 2053
Configuring WebSphere Portal to use Oracle RAC ............................................. 2055
Installing Oracle RAC ............................................................................................ 2056
Create users ........................................................................................................... 2057
Creating databases ................................................................................................ 2059
Modifying database properties ............................................................................ 2060
Setting up databases ............................................................................................. 2063
Creating JCR table spaces ..................................................................................... 2064
Assigning custom table spaces ............................................................................ 2066
Configure WebSphere Portal to use Oracle RAC ................................................ 2068
Configuring WebSphere Portal to use SQL Server ............................................. 2071
Welcome to the WebSphere Portal Version 6.1.5 documentation for Portal Server, Enable and Extend. In this documentation you will find information about planning your installation, installing WebSphere Portal, configuring databases, user registries, security, and SSO, administering portal sites, developing portlets, troubleshooting, and much more.

Getting started
Product overview
What's new
Planning for WebSphere Portal
Exploring the sample site templates
Installing WebSphere Portal
Configuring WebSphere Portal
System requirements

Common tasks
Administering WebSphere Portal
Personalizing your content
Integrating with WebSphere Process Server
Integrating with collaboration software
Integrating mashups
Working with composite applications
Managing Web content
Designing a portal site
Developing portlets

Troubleshooting and support
Troubleshooting WebSphere Portal
Support for WebSphere Portal
Downloads

More information
WebSphere Portal home page
IBM Collaboration Solutions Catalog
Resources for administrators and developers
WebSphere Portal Family Wiki
WebSphere Portal Training
WebSphere Portal zone on developerWorks
Redbooks home page
Product overview

IBM® WebSphere® Portal consists of middleware applications (called portlets), mashups, and development tools for building and managing secure business-to-business (B2B), business-to-consumer (B2C), and business-to-employee (B2E) portals. Web portals allow partners, employees and customers to choose their user experience, with personalized applications based on role, context, actions, location, preferences and team collaboration needs. IBM WebSphere Portal software provides a composite application or business mashup framework and the advanced tooling needed to build flexible, SOA-based solutions, as well as the unmatched scalability required by any size organization.

The Server offering of WebSphere Portal provides personalization and productivity functions along with the scalable portal framework. IBM WebSphere Portal Server is the foundation offering of the WebSphere Portal product family, with enterprise portal capabilities that enable you to quickly consolidate applications and content into role-based applications, complete with search, personalization, and security capabilities.

The Content Accelerator and Enable offerings add IBM Lotus Web Content Management, document management, enterprise search and enhanced workflow capabilities and the Extend offering includes powerful collaborative features including team collaboration, electronic forms, instant messaging and presence awareness services to enhance portal effectiveness. Additional IBM accelerator offerings provide additional solutions that easily snap on to WebSphere Portal software and can reduce time to value, helping customers reduce the costs of deploying content, automating business processes, collaborating and more.

A portal is a Web site that provides users with a single point of access to Web-based resources by aggregating those resources in one place and by requiring that users log in only to the portal itself, and not to each portlet they use. WebSphere Portal can also deliver Web content to WAP-enabled devices, i-Mode phones, Smart phones, and to various Web browsers. In addition, the IBM Mobile Portal Accelerator multi-channel server and mobile device repository extends portal content dynamically to over 7000 mobile devices, with new updates and devices added as they reach the market.

As an administrator, you can customize WebSphere Portal to meet the needs of your organization, users, and user groups. You can adapt the look and feel of the portal to fit the standards of your organization and to customize page content for users and groups in accordance with business rules and user profiles. Users, such as business partners, customers, or employees, can further customize their own views of the portal. Users can add portlets to pages and arrange them as they want and control portlet color schemes. By aggregating portlets in one place and giving users the power to customize their own desktops, WebSphere Portal gives users a means for doing business efficiently and with high satisfaction.

Portlets are central to WebSphere Portal. As special reusable Java servlets that appear as defined regions on portal pages, portlets provide access to many different applications, services, and Web content. WebSphere Portal ships a rich set of standard portlets, including portlets for displaying syndicated content, transforming XML, and accessing search engines and Web pages. Portlets for accessing Lotus Notes®, IBM Lotus® Domino® and Extended Products, IBM Lotus Sametime®, IBM Lotus Quickr, IBM Lotus Connections, Microsoft Exchange, and instant messaging are included. Several third-party portlets are also available. Examples include Enterprise Resource Planning (ERP), Dashboards, Business Intelligence, Process Management, and Customer Relationship Management (CRM) portlets. In addition, WebSphere Portal ships an API that portlet developers can use to create custom portlets.

WebSphere Portal now includes the Lotus Mashup runtime so you can run widgets inside the portal and even create mashups that consist of both portlets and widgets. Widgets are highly interactive user interface components that are written...
in JavaScript. These widgets are typically very narrow in scope and can easily be created using a script-based language. Widgets can also be a solution for creating a mashup between different backend technologies like a Java EE-based portal server and a PHP-based server.

**WebSphere platform**

WebSphere is IBM's integration software platform. It includes the entire middleware infrastructure -- such as servers, services, and tools--needed to write, run, and monitor 24x7 industrial-strength, on demand Web applications and cross-platform, cross-product solutions. WebSphere provides reliable, flexible, and robust integration software. WebSphere provides software for Service Oriented Architecture (SOA) environments that enables dynamic, interconnected business processes, and delivers highly effective application infrastructures for all business situations. IBM WebSphere Application Server drives business agility by providing millions of developers and IT Architects with an innovative, performance-based foundation to build, reuse, run, integrate and manage Service Oriented Architecture (SOA) applications and services. From business critical and key enterprise-wide applications to the smallest departmental level applications, WebSphere Application Server offers the highest levels of reliability, availability, security and scalability. For additional information about new features, main components, and what each component provides to the overall solution, explore the subtopics of this section.

The following topics provide additional overview information.

- **What's new**
  Learn what's new in IBM WebSphere Portal.

- **Documentation resources**
  The starting point for documentation is the information center. However, there are many sites and resources available to you when working with WebSphere Portal. Knowing where to look for information can save you time and money. Learn more about primary and secondary resources for WebSphere Portal and Lotus Web Content Management documentation and supplemental content.

- **Versatile framework**
  IBM WebSphere Portal provides users a consistent view of portal applications and allows users to define specific sets of applications that are presented in a single context. Depending on the requesting device, the rendering of this application set has to vary to fulfill the requirements of the device.

- **Customization**
  Customizing the user's experience is one of the main goals of IBM WebSphere Portal. User and administrative portlets are provided for customizing content and the look and layout of pages. In addition, tools are provided that allow subject matter experts to personalize content to the needs and interests of each site visitor.

- **Web content**
  IBM Lotus Web Content Management is used to create, manage and deliver content for your Web site. You can create content using the Web content authoring portlet, or create your own customized authoring interface. Web content stored in external content management systems can also be referenced within a Web Content Management system. You can deliver your Web content using Web content viewer portlets, the Web Content Management servlet or pre-render your site to HTML.

- **Streamlined site creation**
  IBM WebSphere Portal comes with pre-built, sample Internet and Intranet Web sites that you can explore to learn about authoring and managing different types of content, or use as a template to streamline development of your own portal. You can also use the New Site Wizard to generate your own portal site.

- **Portlets**
  Portlets are a central part of IBM WebSphere Portal. Portlets are small applications that are independently developed, deployed, managed, and displayed. Administrators and users compose personalized pages by choosing and arranging portlets, resulting in customized Web pages.
**Application Integration**
A portal provides access to content, data, and services that are located throughout the enterprise. These services include predefined connectors and portlets, and tools for creating additional connectors and portlets.

**Collaboration**
Collaboration features in IBM WebSphere Portal are provided through the Domino and Extended Products Portlets (formerly called Lotus Collaboration Center), and through the supporting API called Lotus Collaborative Services. Domino and Extended Products are the Domino Enterprise server and IBM Lotus Sametime (formerly called Lotus Instant Messaging and Web Conferencing).

**Accessibility features**
Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use software products successfully.
What's new
Learn what's new in IBM® WebSphere® Portal.

- IBM WebSphere Portal new features and improvements
  WebSphere Portal includes new features, such as the page builder and WebDAV, as well as enhancements to existing features such as easier page creation and customization and improved security features.

- Portal Search - new key features and capabilities
  Some of the new key features of Portal Search include AJAX for faster search results rendering. Portal Search scope and default behavior can now be customized by adding specialized portlets to the Search Center.

- Lotus Web Content Management new features and Improvements
  IBM Lotus Web Content Management includes new features, such as a new rich text editor and Web content viewers, as well as improvements, such as large file handling, in this release.

- Documentation Improvements and changes
  Documentation improvements in this release build on the iterative improvements introduced in the previous release and are designed to improve portal users’ experience. Don't forget that you can share your thoughts about the documentation with us. Tell us what you think using the Submit Feedback button, available on almost every topic.

Parent topic: Product overview
IBM WebSphere Portal new features and improvements

WebSphere Portal includes new features, such as the page builder and WebDAV, as well as enhancements to existing features such as easier page creation and customization and improved security features.

Easier page creation and customization

Page creation and customization is easier than ever before. Using the new Tab Menu - Page Builder theme, you can create pages, move pages, select pages using tree-like navigation, and change the layout of pages directly from the tab you are on. For this release, the new theme is provided automatically only with a full install. Simply click the Home tab then click Go to Edit Mode, click on More Actions and then click on New Child Page. Click Go to Edit Mode, then click the Customize button to add content or change the page appearance by selecting from a predefined list of styles and layouts. You can also share pages that you created or customized with other users.

Page builder

Page Builder is a new portal feature which provides inline customization capabilities at the user's fingertips, rather than by using the administration portlets. Page builder allows users' authoring experience, as it streamlines the modification of pages by providing editing tools directly on the page. Use the portal page builder feature to quickly and easily create, customize, and share pages with teammates and community members. Add content to your pages using a variety of sources made available on the content catalog: portal portlets and shared pages; IBM Mashup Center feeds and pages; Lotus Connections activities, communities, social bookmarks; and custom content sources that you configure. You can add widgets to mashup pages, but not to other portal pages.

Learn more about portal page builder.

Mashup integration

Mashup integration lets WebSphere Portal users share and incorporate lightweight, dynamic Web applications either by creating a mashup page directly in the portal, or by adding a mashup page from the IBM Mashup Center catalog. Users can also share the mashup pages they create in WebSphere Portal by publishing them to the Mashup Center catalog.

Learn more about mashup integration.

Blog and wiki templates

New blog and wiki template libraries facilitate information sharing and collaboration, making it easier to engage the community. Bloggers can deliver news and commentary and hear from readers who can freely post their own comments and observations. Wikis extend the collaborative environment by supporting shared content creation.

Learn more about blogs and wikis.

New security features

Impersonation allows a user, such as a support specialist, to access a user's workstation to test out a new page, portlet, and so forth, and see issues as they occur on the workstation. Before you can impersonate another user, you first must enable the impersonation feature and assign the Delegator role to the appropriate user.

Learn more about impersonation and how to enable it.

WebDAV

WebDAV for WebSphere Portal allows users to administer portal pages and content of static pages of a portal by using standard operating system tools. This way, client-side administrators and users can browse, read, and write these resources by using file explorers or editors. Portal pages are represented as folders. They can contain subfolders that...
represent child pages. Static pages are placed in a separate folder. Users can work in the folders as usual, for example by performing drag-and-drop operations. The folders also hold property files that contain metadata for portal resources, such as the title and description. Users can edit the property files to update portal resources. When the user saves the updated file, the updates are transferred and applied to the portal directly.

Learn more about WebDAV.

**File store**

Allow widgets and portlets to store files in the Java Content Repository (JCR) via WebDAV. It basically provides two areas to store those files, public or user specific. For the public area, everyone can read, authenticated users can write. For the user specific area (.../users/user_name/...), only the user user_name has access to these files. This area also provides public subdirectory with read-only access (.../users/user_name/public/...).

Learn more about WedDAV file store.

**Analyze user behavior**

New Active Site Analytics enable you to learn more about user behavior on your portal site. You can see which pages they visit and which portlets they use.

Learn more about Active Site Analytics.

**Portal light mode**

Portal now provides a light mode which can improve start up time and reduce memory consumption in production environments.

Learn more about portal light mode.

**Page sharing**

This new feature allow users to share a Portal or Mashup page with another user or group of users. The user receiving the shared page can accept it into their navigation structure.

Learn more about page sharing.

**Template pages**

Portal template pages makes it easy for authors to create new pages. It enables you to portal pages that can be used as templates for creating new pages. All page settings will be carried over when a new page is created using that template.

Learn more about template pages.

**Enhanced feed subscription and presentation capabilities**

The new IBM Syndicated Feed Portlet for WebSphere Portal enables you to integrate, view, and manage RSS and ATOM feeds from your portal pages. You can organize the feeds into new and existing feed categories and extensively customize the presentation style for these feeds. The portlet is supported in a proxied environment and enables dynamic modification of the portlet window title. As an Administrator, you can configure several portlet settings right from within the portlet interface and apply selective locks on a user's ability to customize the portlet. The portlet complies with the Outline Processor Markup Language (OPML) standards for subscription management and supports the export and import of feeds.

Learn more about IBM Syndicated Feed Portlet.

Parent topic: What's new
Portal Search - new key features and capabilities

Some of the new key features of Portal Search include AJAX for faster search results rendering. Portal Search scope and default behavior can now be customized by adding specialized portlets to the Search Center.

The following list gives a brief description of the new key features of Portal Search:

- The Search Center is now based on AJAX to provide faster response times for search results.
- Administrators can customize the Search Center with the following new features:
  - They can configure the Search Scopes as follows:
    - By redefining which scope is the default Search Scope.
    - By redefining what the Search Scope All contains. For example, they can reduce the Search Scope All to return only a subset of search results.
  - They can add two new specialized portlets to the Search Center:
    - **Suggested Links**
      - The Suggested Links portlet displays predefined search results and links to users separately from the regular result set. You can add keywords to indexed documents to control which results appear in the suggested results list. You can configure the portlet to select suggested results based on just the keywords, or on all document content, including keywords.
    - **External Results**
      - When an administrator adds that portlet to Portal Search, the results of a search initiated from the Search page include results from third-party external search engines such as Yahoo or Google. You can add more than one copy of the External Search Results portlet to the Portal Search page, and you can configure each of these to display a specific number of search results.

Parent topic: What's new
Lotus Web Content Management new features and improvements

IBM® Lotus Web Content Management includes new features, such as a new rich text editor and Web content viewers, as well as improvements, such as large file handling, in this release.

New Web content viewers

Lotus Web Content Management includes a new JSR 286 Web content viewer that integrates the portal's page structure with information from a Web content library. As a result, the portal page hierarchy helps to organize the navigation of your content. In addition to leveraging the personalization, presentation, and security benefits afforded by JSR 286, there are additional advantages:

- Web Content pages allow you to attach content directly to a page, so you no longer have to configure a rendering portlet to display content from your content library. Other benefits of the JSR 286 Web content viewer include:
  - Sessions are not required for rendering Web content for both anonymous and authenticated portal users.
  - Performance is improved over the traditional Web content viewer.

The JSR 286 Web content viewer can be configured to create site analysis log entries, enabling you to gather data about how often content is visited.

Portal Search supports the use of seedlists to make crawling Web sites and their metadata more efficient and to provide content owners fine-grained control over how content and metadata are crawled. Support has been added for the new seedlist 1.0 format. With support for seedlist format 1.0 and the JSR 286 Web content viewer, search result links going to content items now display the content items in the context of the portal page where they are rendered. Seedlist format 1.0 is the out-of-the-box seedlist format, however you can configure the portal to use a previous seedlist format if required.

Learn more about the new JSR286 Web content viewer.

New rich text editor

EditLive! is the market leading RTE providing rich desktop/Office tool capabilities like advanced table editing as well as linking to the Web Content Management style sheets, role-based control over the RTE features, accessibility support and more.

Learn how to configure Web Content Management to use the new rich text editor.

New link resources

A new link type is now included to allow you to create links to documents stored in an external ECM or document repository, such as Lotus Quickr. When creating a link to an external document in an HTML or rich text element, you are able to select documents to link to from a pre-defined list of external document repositories.

Learn how to link to ECM content.

New syndication feature

A new tool view been added to the syndication administration to allow you to monitor pending items on a syndicator and manually syndicate those items.

Learn how to syndicate pending items.

Large file handling

Web Content Management now supports documents and other binary files that are greater than 50Mb.
Web content feed integration

The IBM Web Content Integrator is a solution used for integrating externally managed Web content with Web Content Management. Through the use of standard content syndication feed technologies based on RSS 2.0, the Web Content Integrator provides a loosely-coupled mechanism for transferring published content and meta data to the portal after they have been approved in the source system. Once the content and meta data have been transferred to the portal, it is possible to leverage the built-in content management features of Web Content Management to secure, personalize, and display the content to end users.

Working with IBM Web Content Integrator

Enhanced file resource elements

If your file resource element is a file type that can be converted to HTML you can now convert the file to HTML and render the converted HTML directly in your Web content. Examples of supported file types include:

- word processing documents (*.doc, *.odt)
- spreadsheets (*.xls) *
- HTML files (*.htm, *.html)
- Text files (*.txt)

New file upload validation plugin

A new plugin class has been added to allow you to write file validation plug-ins that can be used to validate files uploaded into file resource, image and style sheet elements, and images uploaded into rich text or HTML elements.

- Creating a file upload validation class

Parent topic: What's new
Documentation improvements and changes

Documentation improvements in this release build on the iterative improvements introduced in the previous release and are designed to improve portal users’ experience. Don’t forget that you can share your thoughts about the documentation with us. Tell us what you think using the Submit Feedback button, available on almost every topic.

Restructured instructions for integrating IBM Lotus Domino products

Tasks that are required to integrate IBM® Lotus® Domino®, IBM Lotus Sametime®, and IBM Lotus Quickr with WebSphere Portal now follow a more linear, task-based approach that ensures a more successful outcome.

Glossary

A new glossary explains key terms, definitions, and concepts that portal newcomers are likely to encounter. Information center content has also been reviewed and updated to address inconsistencies in terminology.

Properties files

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the WebSphere Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

More graphics

More topology diagrams were added to illustrate different WebSphere Portal deployments in your network.

Documentation resources

To help you navigate the different information resources, a new topic was added to explain what to expect from the Information Center, Support site and WebSphere Portal Family wiki.
Documentation resources

The starting point for documentation is the information center. However, there are many sites and resources available to you when working with WebSphere Portal. Knowing where to look for information can save you time and money. Learn more about primary and secondary resources for WebSphere Portal and Lotus Web Content Management documentation and supplemental content.

There are three primary sources for content: the WebSphere Portal Information Center, the WebSphere Portal Family wiki, and WebSphere Portal Support site.

Each content source links to the other sources to help you navigate between the resources. Each content source has a specific objective and is intended to be used with the other sources.

The information center is developed to help you take advantage of features based on expected usage patterns and use cases. The wiki content is developed by the community, both inside and outside of IBM, with the intent of sharing experiences with the product and based on actual usage patterns and use cases. The Support content is developed by IBM Support to help you avoid and diagnose issues with the intent of being as responsive as possible.

Information center

The information center includes:
- Overview of the product with new feature highlights, product features, and accessibility
- Planning information for deployment
- Installation instructions targeted to single server for proof-of-concepts or development servers, stand-alone production, and clustered production environments
- Configuration options that are typically done once, or infrequently, and have global affect on the portal
- Administration tasks for day to day usage
- Integration instructions
- Development information to help you develop portlets and composite applications
- Troubleshooting information with logging and tracing information
- Messages to help you diagnose and troubleshoot issues

Key points to remember about the information center:
- It is refreshed periodically based on field feedback
- You can provide feedback directly from each page
- There is one information center for each of the following offerings: WebSphere Portal Enable and Extend, WebSphere Portal Express, WebSphere Portal Enable for z/OS
- It includes links to the Support site technotes database to retrieve the latest technotes for this release

WebSphere Portal Wiki

The wiki includes:
- Supplemental guides, such as the Performance and Tuning Guide
- Best Practices
- IBM Redbooks
- Deployment scenarios
- Multimedia offerings, such as task-based demonstrations
- Reference cards

Key points to remember about the wiki:
- The content is experience driven
- You can add comments and articles
- The wiki is monitored by IBM
- You can subscribe to RSS feeds for new articles, comments, and recent edits.

WebSphere Portal Support page

The Support page includes:
- Technotes written in response to issues with the product or the documentation
- Fix pack downloads including instructions for applying fixes
- Troubleshooting information
- Flashes for high priority issues

Key points to remember about the Support page:
- You can subscribe to updates and new Support content using MyNotifications
- You can download tools, such as the IBM Support Assistant
Versatile framework

IBM® WebSphere® Portal provides users a consistent view of portal applications and allows users to define specific sets of applications that are presented in a single context. Depending on the requesting device, the rendering of this application set has to vary to fulfill the requirements of the device.

The tasks of the aggregation, which are repeated with each request coming from the device, are:

- Gather information about the user, the device, and the selected language.
- Select the active portlets from the set of applications to which the user has access.
- Aggregate the output of the active portlets into a coherent, usable display.

WebSphere Portal also provides the ability to create a custom navigation model, which includes such features as:

- Multilevel navigation
- Customized themes and skins
- Custom navigation - navigation tree can be contributed to by portlets themselves
- Custom arrangement of portlets (and thus content) on a page

Another aspect of the versatile framework is the ability to personalize a user’s portal experience, using "content spots" that render subscribed content based on the user and user’s role in the portal.

Parent topic: Product overview
Customization

Customizing the user’s experience is one of the main goals of IBM® WebSphere® Portal. User and administrative portlets are provided for customizing content and the look and layout of pages. In addition, tools are provided that allow subject matter experts to personalize content to the needs and interests of each site visitor.

Customizing pages

Users can have one or more custom pages and access each one through a different page. A page can contain a group of pages that is organized for a specific purpose. Each page can have a different set of portlets. Depending on authorizations, users can change the look and feel of their pages by using skins and page layouts. Also, page navigation hierarchy is tree-based, allowing any depth of nested pages.

The user or an administrator can set up the contents of each page. Administrators can specify that certain portlets be required, so that users are unable to move them or to remove them from the pages. Each page can have its own color scheme and column layout.

Cascading authorization

An administrator can grant or revoke access to customize a page or portion of a page to other administrators or users. The administrator can determine user's rights to modify a page. Administrators can control the edit authority that other administrators have on a page and its contents. This is designed to help organizations enforce policies and consistency, and create region specific portals with some centrally managed content. This control is best explained through an example.

The first administrator can determine that a page will have three columns and not allow the column layout to be modified by any other administrators. A second administrator with lesser access cannot modify the column layout but can add portlets to these columns. The following figure shows a page split into three columns. Administrators can add portlets to these columns.

The second administrator adds a stock portlet to column one and a company news portlet to column two. This administrator wants these portlets to be available to everyone and does not want them to be removed. However, the
administrator can add portlets to the columns. Therefore, the portlets are locked and cannot be removed by other administrators with lesser access. The following figure shows an example of how cascading authorization from one administrator to another would look.

**Skins and themes**

WebSphere Portal uses Java Server Page (JSP) templates, cascading style sheets, and images to define the look of pages. You can modify these templates to control visual aspects of WebSphere Portal, perhaps to add company-specific brand elements or to achieve a different color scheme and visual style. The system for defining color themes and skins supports multiple skins per theme, additional branding elements, navigation styles, and dynamic, browser-independent cascading style sheets.

You can apply skins and themes to a page, not only to the overall product. You can apply different skins individually to portlets as well, so that the appearance of a portal can be fine-tuned to meet any user need. By using a different theme for each page, a single installation can give the appearance of supporting many virtual portals.
Branding elements
You can change all visual elements of WebSphere Portal, including the masthead, the navigation areas, graphics, portlet title areas, and style sheets, to give WebSphere Portal a custom look. You can use standard file formats, such as JPEG, GIF, CSS, and JSP files, to define the look and layout.
The structure of the component installation folder contains folders named "skin" and "theme," with folders "html," "wml," and "chtml." These folders contain most of the files that are used for defining the basic structure of the home page, its color schemes, and portlet decorations. Portal designers can copy these folders and modify their contents to create a custom look and feel. The theme administration portlet registers the new files.

Changing portlet layout
You can change the placement of individual portlets on a page by using the drag-and-drop feature. To rearrange a portlet on a page, click the title bar of the portlet and then drag the portlet to a new location on the page. You can also add portlets to the page for quick and easy page customization by dragging portlets from the Portlet Palette to the page.

Personalization
The Personalization component selects content for users, based on information in their profiles and on business logic. With Personalization facilities, subject matter experts can select content that is suited to the needs and interests of each site visitor. These Web-based tools help companies quickly and easily leverage content that is created by business and subject matter experts. Personalization involves three basic personalization components:
- User Profile: information about users of the site, including user attributes
- Content Model: defines attributes about content, such as product descriptions, articles, and other information
- Matching Technology: engines that match users to the right content; includes filtering, rules, recommendation engines, or combinations of all three.
The Personalization and WebSphere Portal components share a common user profile and content model. The model is based on the WebSphere resource framework interfaces classes. This means that personalization rules can easily be added to portlets to select content and target it to WebSphere Portal registered users.
Personalization classifies site visitors into segments and then targets relevant content to each segment. Business experts create the rules for classifying users and selecting content, using Web-based tools.
Personalization also includes a recommendation engine that provides collaborative filtering capabilities. Collaborative filtering uses statistical techniques to identify groups of users with similar interests or behaviors. Inferences can be made about what a particular user might be interested in, based on the interests of the other members of the group.
New campaign management tools are also included with Personalization. Campaigns are sets of business rules that work together to accomplish a business objective. For example, an HR manager might want to run a campaign to encourage employees to enroll in a stock purchase plan. The HR manager would define a set of rules that are shown to accomplish this business objective. Campaigns have start and stop dates and times and can be e-mail- and Web-page based. Several campaigns can run simultaneously and can be prioritized.
Implicit profiling services can collect real-time information about site visitor actions and then construct personalization business rules using this data. To analyze the effectiveness of the site and its personalization strategies, the server provides reports for the business owner of the site. This helps the company measure the effectiveness of the business rules and campaigns in achieving its objectives.

Universal access
The system of page templates, themes, skins, and portlet rendering is fully enabled for internationalization and accessibility by people with disabilities. For globally accessible portals, WebSphere Portal searches for and selects the proper JSP pages, based on the target browser and its settings for language and country.
Web content

IBM® Lotus Web Content Management is used to create, manage and deliver content for your Web site. You can create content using the Web content authoring portlet, or create your own customized authoring interface. Web content stored in external content management systems can also be referenced within a Web Content Management system. You can deliver your Web content using Web content viewer portlets, the Web Content Management servlet or pre-render your site to HTML.

Design and content separation

In a Web Content Management system, the design and layout of your Web site is separated from the content of your Web site. This allows you to make updates to the look and feel of your Web site without having to revisit your content. Conversely, changes to your content do not require updates to your layout and design.

- The content of your Web site is stored as elements in either context specific site areas or content items, or as reusable components.
- Some element-types store static pieces of HTML or text, others dynamically generate Web page features such as navigation.
- The layout of the pages in your Web site are defined in presentation templates using HTML.
- You specify which elements or components you want to display within each presentation template using Web Content Management tags.

Web content libraries

The items that make up your Web Content Management system are stored in Web content libraries. Your Web content management system can contain multiple libraries. The number of libraries required is determined by the type of Web site you are creating, and the types of users who require access to each library. Separating your site into these libraries enables you to better control the access to each library, and also allows you to setup different syndication strategies for each library.

Web content authoring systems

One of the primary uses of Web Content Management is to build Web content authoring systems for different users:

- The default authoring interface used by Web Content Management is the Web content authoring portlet. You can use separate authoring portlets for different users and groups allowing you to customize the authoring experience for different types of users.
- The inline editing features of Web Content Management can be used to allow users to create and edit content directly in the rendered Web site. This is useful when delivering an intranet solution or a Wiki-style Web site.
- The Web Content Management API can also be used to create custom authoring interfaces.

Preinstalled Web content libraries

A set of preinstalled Web content libraries are supplied to allow you to add blog and wiki features to your Web sites. Use blogs, blog libraries, and wikis to tap into the power of the community and to change the way you work.

- Working with preinstalled Web content

Accessing external content
The Web content used in a Web Content Management system can also be stored and managed in external content management systems. The Web Content Integrator is used to import content into a Web Content Management system using the RSS feed format. Content from federated content systems can be linked directly into a Web Content Management system and inserted in element designs. You can use WebDAV to import content from a file system into a Web Content Management system. You can also use Web Content Management to create rules that can access content from external sources.

- Working with IBM Web Content Integrator
- Using WebDAV with Web content
- Setting up support for federated documents
- Inserting a link to ECM content
- Using rules

Content management features
A range of content management features are available within Web Content Management:
- You can create workflows to control the verification and eventual approval of the various item types that make up a Web Content Management system. Only if an item is approved at all stages up to a published stage can it be viewed on your Web site.
- You can save versions of items that can be restored at a later date.

- Item management features

Web site delivery
Web content can be delivered to your Web site viewers using Web Content Viewer portlets, as pre-rendered HTML files, or as a standard Web site delivered using a servlet. The type of delivery method you use to delivery Web content you your viewers will depend on the type of content being delivered, and the type of viewers your Web site is intended for.
- Standard Web site delivery should be used when you don't need to use any WebSphere Portal based features such as authoring tools.
- Web content viewers are portlets that display content from a Web content library as part of a portal page. If your presentation is simple, a single Web content viewer can be sufficient, but you can also use multiple Web content viewers to aggregate content from different libraries and provide a richer experience for your users.
- You can pre-render a complete Web site into HTML and save it to disk. The pre-rendered site can then be used as your live site and displayed to end users using either Web Content Management or a Web server. You deploy a pre-rendered site when you are not using any WebSphere Portal features and your content is static and is only updated periodically.

- Web content delivery methods

Extending Web Content Management
You can extend the standard features of Web Content Management using the Web Content Management API. You can use the API to develop:
- JSP content that can be stored in JSP elements that can be referenced within presentation templates and element designs.
- Custom Web Content Management plugins that can be added as enterprise applications in your system. You can use these to develop custom workflow actions or create file upload validation rules.
Streamlined site creation

IBM® WebSphere® Portal comes with pre-built, sample Internet and Intranet Web sites that you can explore to learn about authoring and managing different types of content, or use as a template to streamline development of your own portal. You can also use the New Site Wizard to generate your own portal site.

Sample sites

The sample Intranet and Internet sites are not automatically installed when you install WebSphere Portal, but you can add them by running the configure-express task. For more information, see the topic that describes installing WebSphere Portal on your particular platform. For example, if your portal server is running on Microsoft Windows, see Installing WebSphere Portal on Windows.

You can access the sample intranet and internet sites through the More menu in the main portal banner, or by opening the appropriate URL in a browser:
- Intranet sample site: http://hostname.example.com:10040/wps/portal/intranet
- Internet sample site: http://hostname.example.com:10040/wps/portal/internet

Use the Intranet site template as a starting point for creating your own employee site. Home, Work, and Resources pages come seeded with placeholder content that you can use as is or customize. Easy to use list portlets let you organize and manage announcements, news, events, FAQs, and links.

Use the Internet site template to get a jumpstart on building a Web site where customers can learn about product offerings, marketing promotions, and company news.
Creating portal sites on demand

Use the **New Site Wizard** to generate your own site – without needing any portal development skills or assistance from an administrator. Just select a site template from the available samples, choose the look and feel you want, and then let the wizard do the rest.

The wizard automatically creates new sites as virtual portals; however, administrators and developers can extend the wizard to create any type of portal site. Portal developers can also enhance the **New Site Wizard** by creating custom site templates, and then adding them to the wizard.

Download the **New Site Wizard** from the IBM WebSphere Portal Business Solutions Catalog. Package contents include the portlet .war (Web Application Archive) file, supporting files and directories, and detailed instructions in a .pdf file. After deploying the wizard, you need to add the portlet to a page that users can access. For more information, see the .pdf file that comes with the wizard.
Parent topic: Product overview

Related Information

- IBM WebSphere Portal Business Solutions Catalog
- IBM New Site Wizard
Portlets

Portlets are a central part of IBM® WebSphere® Portal. Portlets are small applications that are independently developed, deployed, managed, and displayed. Administrators and users compose personalized pages by choosing and arranging portlets, resulting in customized Web pages.

WebSphere Portal ships a rich set of standard portlets. For the most up-to-date information about portlets, including the latest portlets that are available for download, visit the IBM WebSphere Portal Business Solutions Catalog. Or, refer to Developing portlets for information on creating custom portlets.

Portlet applications

Portlets are more than simple views of existing Web content. A portlet is a complete application, following a standard model-view-controller design. Portlets have multiple states and view modes, plus event and messaging capabilities.

Portlets run inside the application server, similar to the way a servlet runs on an application server, but are aggregated to a complete Web page by the WebSphere Portal server. The portlet container provides a run-time environment where portlets are instantiated, used, and finally destroyed. Portlets rely on the WebSphere Portal infrastructure to access user profile information, participate in window and action events, communicate with other portlets, access remote content, look up credentials, and store persistent data.

Generally, portlets are administered more dynamically than servlets. For example, portlet applications consisting of several portlets can be installed or removed while the WebSphere Portal component is running. The settings and access rights of a portlet can be changed by an administrator while WebSphere Portal is running, even in a production environment.

Portlet modes allow a portlet to display a different user interface, depending on the task that is required of the portlet. A portlet has several modes of display that can be invoked by icons on the portlet title bar: View, Help, Edit, Configure, and Edit Shared Settings.

A portlet is initially displayed in View mode. As the user interacts with the portlet, the portlet can display a sequence of view states, such as forms and responses, error messages, and other application-specific states. Help mode provides user assistance. Edit mode lets the user change portlet settings. For example, a weather portlet might provide an Edit page for users to specify location. Users must be logged in to WebSphere Portal to access Edit mode. Configure mode changes the default look of the portlet for all portlet instances and Edit Shared Settings changes the look of the portlet on a specific page.

Each portlet mode can be displayed in normal, maximized, or minimized state. When a portlet is maximized, it is displayed in the entire body of a page, replacing the view of other portlets. When a portlet is minimized, by default, only the portlet title bar is displayed on the page.

Portlet API

The Java Portlet Specification addresses the requirements of aggregation, personalization, presentation, and security for portlets running in a portal environment. WebSphere Portal supports both portlet standards JSR 168 and JSR 286. For more information about the standard portlet API, refer to the topic about the Standard portlet API.

Portlet communications

WebSphere Portal allows portlets to communicate with each other. Portlet communication can be used to exchange data between portlets. This makes the portal easier to use.

The portal supports events as defined in the JSR 286 specification. It allows administrators to wire portlets by using the portal user interface.

For example, one portlet can display information about accounts, and a second portlet displays information about
transactions that have occurred for one of the accounts over the last 30 days. To do this, the transactions portlet needs to obtain the appropriate account information when it displays the transaction details. This is accomplished by communication between the two portlets, using events as described in the JSR 286 specification. In this example, the account portlet defines a publishing event in its portlet descriptor. The transaction portlet defines this event as a processing event in its portlet descriptor. By using the portal user interface, you can now wire those two portlets together. After you did this, when the account portlet throws an event, the transaction portlet receives this event and can show information about the transactions of this account.

**Portlet services**

Portlet services are used to provide common functionality to portlets. Each portlet service has its own service specific interface for the functionality that it offers. WebSphere Portal supports portlet services for standard portlets. Standard portlets use a JNDI lookup to retrieve a PortletServiceHome object, which is used to retrieve a portlet service implementation. A portlet service can be invoked only by the code within a portlet. For more information about portlet services in the portal, refer to the topic about Portlet services.

**Creating and customizing portlet applications**

WebSphere Portlet Factory is included with WebSphere Portal and provides a robust selection of builders that supercharge the portlet development process without writing code. WebSphere Portlet Factory’s rapid application development technology enables portlet creation 40 – 70 percent faster than using traditional J2EE development methods. With WebSphere Portlet Factory you can rapidly develop and deploy custom service-oriented portlets and rich, interactive Web 2.0 style applications with features like drag-and-drop, in-line editing, type-ahead search and intelligent page refresh functionality. WebSphere Portlet Factory transforms operational data into high-value business information by integrating data from a wide variety of packaged enterprise applications, repositories and data sources including SAP, Siebel, PeopleSoft, Lotus Domino, Web and REST services and leading relational databases via a rich, pre-built connector library. Native WebSphere Portal integration enables creation of composite applications with embedded collaboration features that facilitate real-time problem resolution.

Using WebSphere Portlet Factory's patented dynamic profiling functionality, developers can easily empower business-user led portlet customization via personalization and create dynamic, micro-targeted applications that vary portlet content based on user role, geography and more. Applications built with WebSphere Portlet Factory can be deployed to multiple run-time environments to provide the right user experience based on target audience, including WebSphere Portal, Mashup Center, Lotus Notes® and Expeditor and WebSphere Application Server. WebSphere Portlet Factory applications are standards based and comply with portlet standards including JSR 168 and JSR 286.
Application integration

A portal provides access to content, data, and services that are located throughout the enterprise. These services include predefined connectors and portlets, and tools for creating additional connectors and portlets.

Enterprise resource planning (ERP) and customer relationship management (CRM) systems are excellent candidates for portlets because efficient, personalized access to these functions provides measurable return on your portal investment.

IBM provides connectors to enterprise applications using the Java Connector Architecture (JCA).

**Standard Java connectors**

JCA is a standard architecture for integrating Java 2 Enterprise Edition (J2EE) applications with enterprise information systems that are not relational databases. Each of these systems provides native APIs for identifying a function to call, specifying its input data, and processing its output data. The goal of the JCA is to provide an independent API for coding these functions.

JCA also defines a standard Service Provider Interface (SPI) for integrating the transaction, security, and connection management facilities of an application server with those of a transactional resource manager. Thus, JCA is a standards-based approach to managing connections, transactions, and secure access to enterprise application systems. IBM JCA connectors provide access to systems such as SAP, PeopleSoft, J.D. Edwards, Oracle Enterprise Edition, CICS, IMS, and Host-on-Demand. Leveraging its CrossWorlds acquisition, IBM plans to develop and integrate JCA connectors to many other systems.

Rational Application Developer provides a complete development and unit test environment for applications that use JCA connectors, Web services, and microflows. Rational Application Developer tools include support for Web Service Definition Language (WSDL), developer versions of the connectors, the Web Services Invocation Framework (WSIF), and the microflow engine.

Parent topic: Product overview
Collaboration

Collaboration features in IBM® WebSphere® Portal are provided through the Domino® and Extended Products Portlets (formerly called Lotus Collaboration Center), and through the supporting API called Lotus® Collaborative Services. Domino and Extended Products are the Domino Enterprise server and IBM Lotus Sametime® (formerly called Lotus Instant Messaging and Web Conferencing).

Although it had previously been, and still is, possible to access the functionality of Lotus Sametime using only the Collaborative Services API components within WebSphere Portal, the Domino and Extended Products Portlets extend the capabilities beyond simple integration and access. Portlets are the mechanism by which the Domino server products are fully integrated into WebSphere Portal. Domino and Extended Products Portlets provide an online directory with people awareness, and integrated tools for managing online meetings. All these collaboration features help people in your organization work together and share information online to achieve their business goals. A collaborative portal can improve your organization's responsiveness, innovation, competencies, and efficiency.

WebSphere Portal supports Domino and Lotus Sametime, bridging to the world of J2EE and open standards, and extending Domino's capabilities by providing the means to connect different sources of data, regardless of vendor. For information on supported versions of these products, see the WebSphere Portal detailed system requirements.

Configuration of the Domino and Extended Products and portlets is automated by the Domino-Portal Integration Wizard. The wizard saves many manual steps for the administrator.

Collaborative Portlets

Examples of all the Domino and Extended Products Portlets are supplied on the Collaboration page and on the Messaging page. The portlets are installed with WebSphere Portal and deployed automatically, requiring of the administrator only a number of server and other configuration tasks before users can take advantage of a suite of integrated collaborative applications.

Domino and Extended Products Portlets include:
- Domino Web Access
- Sametime Contact List
- Lotus Notes View
- People Finder
- Who Is Here

Domino and Extended Products

You can enhance the collaborative features of WebSphere Portal, and add features such as people awareness to collaborative portlets, by choosing to install and enable any of the following Domino and Extended Products:
- IBM Lotus Sametime adds the online status indicator to person links, and adds actions to the Person menu.
- IBM Lotus Domino provides access, from many collaborative portlets, to Lotus Notes discussion and teamroom databases.

Lotus Sametime features

When Lotus Sametime is enabled in your portal configuration, users can work with the complete set of people awareness functionality, which includes instant messaging and application sharing through e-meetings. Person names appear aware - with a dynamic online status indicator. Move the cursor over an active (underlined) person's name to see the Click for Person Card option, and then click Click for Person Card to see the Person menu. The Person menu can include the following options:
You might choose not to enable Lotus Sametime in your portal configuration. If Lotus Sametime is not enabled, then people awareness functionality will be restricted to the following features:

- People's names appear as hyperlinks, but with no people awareness icon next to each name.
- Moving the cursor over an active (underlined) person's name displays the **Click for Person Card** option which you can click to display the Person menu. Menu options will be limited to actions such as **Show Profile**, that are native to WebSphere Portal.

**Lotus Domino features**

When a Domino server is available, portal users have access, using the Lotus Notes View (formerly "Notes and Domino") portlet, to any Lotus Notes database on the server. And other portlets, such as Domino Web Access, use a Domino server configured for use in the portal.

**Collaborative Services APIs**

Lotus Collaborative Services are Java APIs that provide the building blocks for integrating the functionality of Domino and Lotus Sametime into the portal and portlets. Using Collaborative Services, application developers can design and implement user interface extensions in portals and portlets that extend the functionality of collaborative portlets.

The primary goal of Collaborative Services is to provide the data for portlets' user interface and to allow developers to execute actions on installed Domino and Extended Products. Collaborative Services include no platform-specific code, hide the configuration details of Domino and Extended Products installed in the enterprise, and (except for the Person tag) are user-interface neutral. Application developers can add collaborative functionality to a portlet without having to know the details of server configuration and with total control of user interface design and implementation. These benefits make Collaborative Services effective for implementing mobile applications.

**Parent topic:** Product overview

**Related concepts**

- Integrating with collaboration software
- People awareness using the Person tag
Accessibility features

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use software products successfully.

This version of IBM® WebSphere® Portal:
- Supports installation through a command line interface known as console mode. This is the accessible equivalent of installing by using the graphical user interface.
- Supports interfaces commonly used by screen readers and screen magnifiers (Windows only)
- Supports use of screen-reader software and digital speech synthesizers to hear what is displayed on the screen.
- Can be operated using only the keyboard
- Allows the user to request more time to complete timed responses
- Supports customization of display attributes such as color, contrast, and font size
- Communicates all information independently of color
- Supports the attachment of alternative input and output devices
- Supports alternatives to audio information
- Supports adjustable volume control
- Does not flash the screen at rates that could induce epileptic seizures
- Provides documentation in an accessible format

Note: For best results when using a screen reader to view WebSphere Portal, use Freedom Scientific JAWS 10 or higher and Firefox 3 or higher.

The documentation includes the following features to aid accessibility:
- All documentation is available in HTML formats to give the maximum opportunity for users to apply screen-reader software technology.
- All images in the documentation are provided with alternative text so that users with vision impairments can understand the contents of the images.

When appropriate, the documentation for specific product features contains additional information about accessibility. See the IBM Human Ability and Accessibility Center for more information about the commitment that IBM has to accessibility:

Parent topic: Product overview

Related concepts
Accessibility and keyboard shortcuts in the information center
Planning for WebSphere Portal

Before deploying IBM® WebSphere® Portal, it is important to understand requirements and configuration considerations for high availability and failover. Become familiar with the system and database architecture and security considerations and plan your deployment. You can also find useful information in the WebSphere Portal Family wiki about best practices to help you plan your WebSphere Portal environment.

- **System requirements**
  
  Before installing IBM WebSphere Portal, review the hardware and software requirements to ensure you have the supported versions of prerequisite and corequisite software as well as the required hardware.

- **WebSphere Portal Support Statement**
  
  This support statement proposes a revision to the definition of "supported" and "unsupported" with respect to the various products of which IBM WebSphere Portal depends on for proper operation.

- **User IDs and passwords**
  
  Understanding character limitations for user IDs and passwords is important because they are used throughout the system to provide access and secure content. The character limitations provided here apply to the IBM WebSphere Portal administrator, IBM WebSphere Application Server administrator, database administrator, LDAP server administrator, and user IDs. Database and LDAP servers can have more restrictive limitations than provided here. Therefore you should check the database and LDAP server product documentation for restrictions. Failure to correctly define user IDs and passwords during the installation process can result in installation failure. In addition, your company may have more restrictive user ID and password requirements that you must also follow.

- **Preparing your operating system**
  
  Each operating has unique preparation needs to ensure a successful installation. Before you continue with the installation wizard, make sure you have properly prepared your operating system and that you have all the required hardware and software prerequisites and corequisites.

- **Server topologies**
  
  Topology diagrams help you visualize different configurations that you can setup to support various user and system load requirements. The topology diagrams are representative of basic configurations.

- **Web Content Management environments**
  
  To use a Web Content Management system, you will need to use a set of Web Content Management servers.

- **Web servers**
  
  By default IBM WebSphere Portal uses the internal HTTP transport within IBM WebSphere Application Server to handle requests. However, because WebSphere Application Server also supports the use of an external Web server, you can access WebSphere Portal from your Web server. You can use a local Web server on the same machine as WebSphere Portal or you can use a remote Web server on a different machine. A remote Web server is typical for a production environment or other high-traffic configuration and is also typically placed in demilitarized zones (DMZ) outside a firewall to protect portal ports.

- **Database considerations**
  
  A simple development environment can rely on the out-of-box database configuration using Apache Derby. Installing with Derby lets you quickly get IBM WebSphere Portal installed and running in a proof-of-concept environment. For a
production environment, WebSphere Portal supports IBM DB2 Universal Database™ Enterprise Server Edition.

- **User registry considerations**  
  A user registry or repository authenticates a user and retrieves information about users and groups to perform security-related functions, including authentication and authorization.

- **Security and authentication considerations**  
  Security and authentication are key elements of a production environment. Learn about single sign-on, credential vaults and external security managers.

- **Cluster considerations**  
  To increase capacity and availability, multiple portal servers can be clustered using IBM WebSphere Application Server Network Deployment. In a cluster the portals share a common configuration and the load is distributed evenly across all cluster instances.

- **WebSphere Process Server integration**  
  IBM WebSphere Portal integrates business processes and related tasks that are managed by IBM WebSphere Process Server. Using process integration, you can manage human tasks for business applications within the portal environment to support service-oriented architecture (SOA). Portals that provide the interface for human tasks in business processes that are managed by a process server are referred to as process portals. Successful deployment of process portals requires planning for installation, configuration, and use within your enterprise.

- **Release notes**  
  Known issues and problems are centrally available on the support page. Links into the support knowledge base are integrated throughout the information center to make sure you have the most current information. Before you start the installation process, check the IBM Support site for the most current information about known limitations or issues. Use the following dynamic queries to find late breaking information about this release.

**Related information**  
- WebSphere Portal Family wiki: Best practices
System requirements

Before installing IBM® WebSphere® Portal, review the hardware and software requirements to ensure you have the supported versions of prerequisite and corequisite software as well as the required hardware.

See the detailed system requirements document at the following URL:
http://www.ibm.com/support/docview.wss?rs=688&uid=swg27007791

**Parent topic:** Planning for WebSphere Portal
WebSphere Portal Support Statement

This support statement proposes a revision to the definition of "supported" and "unsupported" with respect to the various products of which IBM® WebSphere® Portal depends on for proper operation.

Introduction

WebSphere Portal requires the use of several collateral products for its normal operations. In particular, it requires WebSphere Application Server, a database, a repository for user information (typically an LDAP), and other products depending on specific customer requirements.

During the testing of a new release, Development generally tests WebSphere Portal with a prescribed list of these collateral products. These products are designated as "Supported Products" in the documented hardware and software requirements for that release.

Because the list of "Supported Products" cannot reasonably describe all possible configurations that a customer may need to use, some customers have voiced concerns about the level of support that will be provided for configurations that are not specifically designated as "Supported." This document is intended to provide clarification of the level of support that can be generally expected for the current release with various combinations of dependent products.

Note: Although the statements in this document reflect the general level of support that can be expected for WebSphere Portal, the terms and conditions of any specific support offering, license or other Agreement you might have with IBM will determine the actual delivered support for the product. Nothing herein shall be construed as supplementing, modifying or superseding the terms of your IBM license agreement for WebSphere Portal or any other agreement you might have with IBM, nor shall it create any obligation for IBM to deliver a level of support other than might be set forth in such Agreements.

Categories of Support

There are three (3) categories of support for collateral products to WebSphere Portal. They are "Supported Products", "Newer Versions and Releases of Supported Products" and "Unsupported Products". The definition and support statement for each category follows:

- **Supported Product**

  A "Supported Product" is a product (at a specified version, release and fix level) that was tested by Development and is known to work with WebSphere Portal.

  Products in this category are supported as per the terms of your WebSphere Portal License Agreement. PMRs (Problem Management Records) will be accepted by IBM Support in accordance with the conditions of the WebSphere Portal License Agreement.

- **Newer Versions and Releases of Supported Products**

  Many products outside the specific version(s), release(s) or fix pack(s) of the "Supported" version, (referenced in the documented hardware and software requirements) may not have been explicitly tested by IBM WebSphere Development, yet can reasonably be expected to perform within the accepted bounds of reliability, function and performance by a customer.

  Products that fall into this category are typically newer releases or fix levels of a product already in the "Supported Product" category or a product that adheres to a standard API that WebSphere Portal supports (such as an LDAP server). Some specific examples might include a newer operating system fix level, a WebSphere Application Server (WAS) fix pack newer than the original "Supported" fix pack level, an IBM Java (JVM) fix pack, a new fix pack or release of DB2 or an updated LDAP server.

  For products that fall into this category, support is as follows:

  *For IBM products, such as IBM Directory Server or Domino LDAP, IBM DB2, IBM JDKs (JVMs) and WebSphere Application Server, WebSphere Portal will fully support fix-pack, release and version updates that do not significantly change interfaces or other underlying support that WebSphere Portal depends on for its functionality. If and when a newer
release of one of these products is shipped that WebSphere Portal cannot accommodate, that fact will be noted as described in the next section entitled “Unsupported Products”. Note that in order for WebSphere Portal to support an update to a database or LDAP product, WebSphere Application Server must support that update as well.

For non-IBM products, the Support team will make a commercially reasonable effort to support products in this category. Support will accept problem reports (PMRs) for the appropriate releases using these untested products. If Support is able to recreate the reported problem using a "Supported" version of the product, we will attempt to fix the problem. If Support is not able to recreate the problem with a "Supported" version of the product in question and is not able to resolve the problem on the untested version of the product in question, Support will look to the support organization for the product in question to provide resolution. Please note that varying degrees of customer involvement may be necessary to handle this process for non-IBM products.

If the support organization for the untested product in question is unable to resolve the problem, Support will deem that version, release or fix pack level of the untested product in question to now be an "Unsupported Product".

- **Unsupported Products**: An "Unsupported Product" is a product (at a specified version, release and fix level) that is known to not work with WebSphere Portal and is therefore not supported. A product can be included in this category as a result of an explicit test effort by Development or as a result of discovery from a prior customer problem. The WebSphere Portal Support team maintains a list, by WebSphere Portal release, of all known "Unsupported Products" published on the Web as a techdoc available to all customers.

WebSphere Application Server has a similar support statement which can be found on the Web.

**Note:** WebSphere Application Server uses specially customized builds of the IBM Java SDKs on certain platforms. Updates to these must be obtained from WebSphere Application Server support.

WebSphere Portal can be sensitive to changes in the underlying WebSphere Application Server. Upgrading to a new fix pack level of the application server is well tolerated and encouraged (such as from WebSphere Application Server version 6.1.0.17 to 6.1.0.19) as long as all required fixes for WebSphere Application Server are available as integrated into that fix pack or by applying an interim fix specifically for that maintenance level. However, upgrading from one version of WebSphere Application Server to the next (such as from 6.0 to 6.1) is quite problematic if not done within the context of a migration of versions and should never be attempted with an "in-place" system. For example, an existing instance of WebSphere Portal version 6.0.1.1 installed and functioning on WebSphere Application Server version 6.0.2.17 cannot be successfully migrated to WebSphere Application Server version 6.1.x simply by using the WebSphere Application Server Migration Tools. Such attempts may result in a non-functional system. Consult IBM WebSphere Portal support for more information on such scenarios, if required.

**Support for LDAP Servers**

LDAP support spans two (2) categories:

- **Fully tested and supported LDAP servers**: The list of fully tested LDAP servers for each release of WebSphere Portal is documented in the detailed system requirements for each release. For more information on system requirements, see the link below. WebSphere Portal support accepts problem reports for the appropriate WebSphere Portal releases using the tested directory servers. These problem reports receive high-priority attention. Features that are tested with these directories include relatively simple search and retrieval functions for user and group objects. Functions outside this scope, such as dynamic groups, referrals, or the Active Directory Global Catalog feature, are considered advanced features and have not been tested with WebSphere Portal. WebSphere Portal support encourages customers to work with their LDAP provider for additional support on these advanced features.

- **Untested and partially supported LDAP servers**: In general, WebSphere Portal support makes a best effort to support directory servers that have not been tested with WebSphere Portal. WebSphere Portal support accepts problem reports for the appropriate WebSphere Portal releases using untested directory servers. If WebSphere Portal support can recreate the reported problem using a tested LDAP server, staff will attempt to fix the problem. If the support team is not able to recreate the problem on a tested LDAP server, customers are referred to the LDAP provider for further assistance.
Parent topic: Planning for WebSphere Portal

Related Information
- WebSphere Portal Support Statement Addendum - Unsupported products
- Document #7004311, WebSphere Application Server has grown and so have your requirements
- WebSphere Portal detailed system requirements
User IDs and passwords

Understanding character limitations for user IDs and passwords is important because they are used throughout the system to provide access and secure content. The character limitations provided here apply to the IBM® WebSphere® Portal administrator, IBM WebSphere Application Server administrator, database administrator, LDAP server administrator, and user IDs. Database and LDAP servers can have more restrictive limitations than provided here. Therefore you should check the database and LDAP server product documentation for restrictions. Failure to correctly define user IDs and passwords during the installation process can result in installation failure. In addition, your company may have more restrictive user ID and password requirements that you must also follow.

When a person signs up as a user or when an administrator enrolls a user, they must complete the user information form. On this form, do not enter characters that might not be supported. Regardless of what characters you are able to enter on the user information form, user ID and passwords are limited to the valid characters described here. You can specify other characters in the First Name and Last Name fields. If your company policy is more restrictive, you can provide that information to your users in the enrollment form help or as in-line help directly on the form. Important: WebSphere Portal cannot create user IDs or passwords that contain spaces, although it fully supports any existing user IDs and passwords or those created in the user repository that contain spaces.

Under normal circumstances a valid user ID and password can contain the following characters: Note: The only supported characters in IBM i5/OS are lower case characters, upper case characters, numbers, and the underscore.

- Lower case characters {a-z}
- Upper case characters {A-Z}
- Numbers {0-9}
- Exclamation point {!}
- Open parenthesis {(}
- Close parenthesis {)}
- Dash {-}; this character is not supported as the first character in the user ID or password
- Period {.}; this character is not supported as the first character in the user ID or password
- Question mark {?}
- Open bracket {[}
- Close bracket {]}
- Underscore {_}; this is the only supported special character in i5/OS
- Grave accent {'}
- Tilde (\~)

Important: These are all ASCII characters. Non-ASCII characters are not allowed for username or password.

Note: (UNIX only) Some tasks may require you to enter the fully qualified user ID. If your fully qualified user ID contains a space; for example: cn=wpsadmin, cn=users, l=SharedLDAP, c=US, ou=Lotus, o=Software Group, dc=ibm, dc=com, you must place the fully qualified user ID in the properties file or into a parent properties file instead of as a flag on the command line. For example, create a parent properties file called mysecurity.properties, enter the fully qualified user ID, and then run the task: ./ConfigEngine.sh task_name -DparentProperties=/opt/mysecurity.properties.

Note: (Windows only) Some tasks may require you to enter the fully qualified user ID. If your fully qualified user ID contains a space; for example: cn=wpsadmin, cn=users, l=SharedLDAP, c=US, ou=Lotus, o=Software Group, dc=ibm, dc=com, you must place quotes around the fully qualified user ID before running the task; for example,
The table below contains a list of the required fields on the user information form and the supported characters.

Table 1. Valid characters and unsupported characters for user information

<table>
<thead>
<tr>
<th>User information</th>
<th>Valid characters</th>
<th>Unsupported characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ID</td>
<td><strong>Note:</strong> The only supported characters in IBM i5/OS are lower case characters, upper case characters, numbers, and the underscore. Lower case characters (a-z) Upper case characters (A-Z) Numbers (0-9) Exclamation point (!) Open parenthesis (()) Close parenthesis () Dash (-); this character is not supported as the first character in the user ID or password Period (;); this character is not supported as the first character in the user ID or password Question mark (?) Open bracket [ ] Close bracket (]) Underscore (_); this is the only supported special character in i5/OS Grave accent (’) Tilde (~)</td>
<td>Only ASCII characters are allowed. <strong>Other restrictions:</strong> The user ID cannot contain spaces; for example, <em>user name</em>. User IDs cannot be longer than 200 characters. If you enter any unsupported characters during the installation, you will receive an error message that states which character is invalid. For example, “The special character [@] was found in the administrative user ID field. Enter the administrative user ID again.” <strong>Important:</strong> You will receive a different error message if you enter any unsupported characters when creating users through the Manage users and groups portlet.</td>
</tr>
<tr>
<td>Password / Confirm password</td>
<td><strong>Note:</strong> The only supported characters in IBM i5/OS are lower case characters, upper case characters, numbers, and the underscore. Lower case characters (a-z) Upper case characters (A-Z) Numbers (0-9) Exclamation point (!) Open parenthesis (()) Close parenthesis () Dash (-); this character is not supported as the first character in the user ID or password Period (;); this character is not supported as the first character in the user ID or password Question mark (?) Open bracket [ ] Close bracket (]) Underscore (_); this is the only supported special character in i5/OS Grave accent (’) Tilde (~)</td>
<td>Diacritics, such as the umlaut, and DBCS characters are not allowed. <strong>Other restrictions:</strong> The password cannot contain spaces; for example, <em>password</em>. Passwords cannot be longer than 128 characters. <strong>Attention:</strong> Login will fail if the password contains any unsupported characters, including DBCS characters. This will happen even if a user is successfully enrolled using a password containing DBCS characters. If you enter any unsupported characters during the installation, you will receive an error message that states which character is invalid. For example, “The special character [@] was found in the password field. Enter the password again.”</td>
</tr>
<tr>
<td>First Name</td>
<td>All characters</td>
<td>n/a</td>
</tr>
<tr>
<td>Last Name</td>
<td>All characters</td>
<td>n/a</td>
</tr>
</tbody>
</table>

**Note:** The above characters are true if the `user.UNIQUEID.charset` parameter is set to `ascii`. If set to `unicode`, the standard Java Letter definition is used and all characters that are recognized as letter or digit by Java are allowed by
default. See the **Puma Validation Service** section in the "Portal configuration services" link below for information about further parameters that can be modified to affect the behavior of Portal's validation of users, groups, and passwords.

**Parent topic:** Planning for WebSphere Portal

**Related reference**

Portal configuration services
Server topologies

Topology diagrams help you visualize different configurations that you can setup to support various user and system load requirements. The topology diagrams are representative of basic configurations.

The single server topology illustrates a simple installation for demonstration, trying the product out, or development environment purposes. The stand-alone topology illustrates a distributed configuration. Clustered topologies illustrate more robust and load intensive hardware configurations and Web Content Management topologies illustrate how different hardware configuration support various authoring and system load requirements. To increase capacity and availability, multiple portal servers can be clustered using IBM® WebSphere® Application Server Network Deployment, where the portals share a common configuration and load is distributed evenly across all cluster instances. The high availability and failover topology illustrates WebSphere Portal in a more complex production environment.

- **Single-server topology**
  
  A single instance of IBM WebSphere Portal provides a fully functional application server and a web site that is capable of serving a medium to small user community. Initially, a single instance is configured to use an internal database based on Derby that is unsuitable for production use and must be replaced by an enterprise class database.

- **Stand-alone server topology**
  
  The stand-alone scenario is different from the single-server since the database server, LDAP server, and Web server software are installed on different physical servers than the IBM WebSphere Portal. This configuration enables you to distribute the software in your network and therefore distribute the processing load.

- **Horizontal cluster topology**
  
  A horizontal cluster consists of at least two nodes with a single cluster instance in each node. Each node in the cluster typically represents a unique IBM WebSphere Portal installation on a unique physical system. Horizontal clusters, like vertical clusters, provide a means of expanding the capacity and availability of the portal site, but unlike vertical clusters, each horizontal cluster instance has a unique set of physical resources (CPU and memory) and thus is not affected by other horizontal instance resource requirements.

- **Single-server topology for Web Content Management**
  
  Organizations that only need a small Web site or intranet site could implement a single server topology for Web Content Management. In a single server configuration, authoring and delivery occur on the same server. This configuration is not recommended for organizations that need a larger Web site or intranet site.

- **Dual-server configuration for Web Content Management**
  
  Organizations that need a large Web site with heavy traffic or that have multiple users authoring content should implement a dual-server configuration. In a dual-server configuration authoring is conducted on one server, or server cluster, and content is delivery to a different server, or server cluster. Authors access and work on the authoring server while Web site users access the delivery server. This reduces the load on each server and allows you to locate your authoring environment behind a firewall.

- **Staging-server topology for Web Content Management**
  
  If you need to deliver large complex Web sites with a large number of site users and content creator, implement a staging server configuration. Implementing a staging server provides an environment for checking for accuracy, issues with the design, and performance. With a staging server configuration authoring, staging and delivery are separated onto different servers. The staging environment can be used for user acceptance testing (UAT) or to allow changes from the authoring environment to accumulate prior to syndicating the changes to the delivery environment in a single batch.
Single-server topology

A single instance of IBM® WebSphere® Portal provides a fully functional application server and a web site that is capable of serving a medium to small user community. Initially, a single instance is configured to use an internal database based on Derby that is unsuitable for production use and must be replaced by an enterprise class database.

Single server deployments are ideal for development, test, demonstration and education, and, as mentioned, some small production sites. A single server, however, presents a single point of failure and must be converted to a cluster or server farm to improve its capacity, availability, and ability to recover from failure conditions.

A Web server configured with IBM WebSphere Application Server’s HTTP plug-in will front a typical server deployment for test or production use. The Web server provides the ability to serve static resources, such as images and style sheets, as well as a plug-in point for a corporate single sign-on (SSO) agent, in the event that WebSphere Portal will participate in a global SSO domain.

The following illustration displays a common topology for a single server environment. Only one server is needed. WebSphere Portal, WebSphere Application Server, and the database are all installed on the same server.

Parent topic: Server topologies

Related concepts

Stand-alone server topology
Horizontal cluster topology
Single-server topology for Web Content Management
Dual-server configuration for Web Content Management
Staging-server topology for Web Content Management
Stand-alone server topology

The stand-alone scenario is different from the single-server since the database server, LDAP server, and Web server software are installed on different physical servers than the IBM® WebSphere® Portal. This configuration enables you to distribute the software in your network and therefore distribute the processing load.

For a stand-alone configuration, you can use an existing, supported database in your network and an existing, supported LDAP directory. You can configure IBM WebSphere Portal to authenticate with the LDAP server. The following illustration displays a common topology for a stand-alone server. The HTTP server, (also referred to as Web server) is installed on a server in a protected network. The LDAP server and database server are also installed on different servers. WebSphere Portal and WebSphere Application Server are installed on the same server.

Parent topic: Server topologies

Related concepts

- Single-server topology
- Horizontal cluster topology
- Single-server topology for Web Content Management
- Dual-server configuration for Web Content Management
- Staging-server topology for Web Content Management
Horizontal cluster topology

A horizontal cluster consists of at least two nodes with a single cluster instance in each node. Each node in the cluster typically represents a unique IBM® WebSphere® Portal installation on a unique physical system. Horizontal clusters, like vertical clusters, provide a means of expanding the capacity and availability of the portal site, but unlike vertical clusters, each horizontal cluster instance has a unique set of physical resources (CPU and memory) and thus is not affected by other horizontal instance resource requirements.

The following diagram illustrates a horizontal cluster. All of the cluster nodes are on individual servers. The nodes are grouped into a cell and managed by a single Deployment Manager. All of the node use a common database and LDAP server. The database and LDAP server could also be clustered and in high traffic networks or networks where failover and recovery are critical, the database and LDAP server would typically be clustered in order to avoid a single point of failure.

- Vertical cluster topology
  A vertical cluster has more than one cluster instance within a node. A node typically represents a single physical server in a managed cell, but it is possible to have more than one node per physical server. It is very simple to add additional vertical clusters to a node, using the Deployment Manager administrative console, as each additional vertical cluster instance replicates the configuration of the first instance; no additional installation or configuration is necessary.

- Combination of horizontal and vertical clusters
  Most large-scale portal sites incorporate a combination of horizontal and vertical clustering to take full advantage of the resources of a single machine before scaling outward to additional machines.

- Multiple clusters
  To improve server availability, failover and disaster recovery, as well as to help span large geographical deployments, it is best to consider deploying multiple portal clusters for the same portal site. With multiple clusters, one cluster can be taken out of production use, upgraded and tested while leaving other clusters in production. This is the best means of achieving 100% availability without requiring maintenance windows. It is also possible to deploy clusters closer to the people they serve, improving the responsiveness of the content they provide.
Parent topic: Server topologies

Related concepts
- Single-server topology
- Stand-alone server topology
- Single-server topology for Web Content Management
- Dual-server configuration for Web Content Management
- Staging-server topology for Web Content Management
- Cluster considerations
**Vertical cluster topology**

A vertical cluster has more than one cluster instance within a node. A node typically represents a single physical server in a managed cell, but it is possible to have more than one node per physical server. It is very simple to add additional vertical clusters to a node, using the Deployment Manager administrative console, as each additional vertical cluster instance replicates the configuration of the first instance; no additional installation or configuration is necessary.

How many vertical instances that can be created in a single node depends on the availability of physical resources in the local system (CPU and memory). Too many vertical cluster instances could exhaust the physical resources of the server, at which point it is appropriate to build horizontal cluster instance to increase capacity if necessary.

The following diagram illustrates a vertical cluster. A single node has multiple instances of IBM® WebSphere® Portal installed. The node is managed by the Deployment Manager. Each instance on the node authenticates with the same LDAP server and store data in the same database server. Although not depicted in the illustration, the database and LDAP servers could also be clustered if needed for failover, increased performance, and high availability.

---

**Parent topic:** [Horizontal cluster topology](#)
Combination of horizontal and vertical clusters

Most large-scale portal sites incorporate a combination of horizontal and vertical clustering to take full advantage of the resources of a single machine before scaling outward to additional machines.

This graphic depicts two horizontal cluster members, node A and B. Both node A and B have 3 vertical cluster members.

**Parent topic:** Horizontal cluster topology
Multiple clusters

To improve server availability, failover and disaster recovery, as well as to help span large geographical deployments, it is best to consider deploying multiple portal clusters for the same portal site. With multiple clusters, one cluster can be taken out of production use, upgraded and tested while leaving other clusters in production. This is the best means of achieving 100% availability without requiring maintenance windows. It is also possible to deploy clusters closer to the people they serve, improving the responsiveness of the content they provide.

When deploying multiple portal clusters, for the most part, each cluster should be seen as a totally isolated system. Each cluster is administered independently and has its own configuration, isolated from the other clusters. This improves the portal topology's ability to be maintained while protecting its high availability. The only exception to this rule is with the sharing of some of the portal database domains, the Community and Customization domains specifically, which are designed to be shared across multiple clusters presenting the same portal site. These domains store portal configuration data owned by the end users themselves, and so it is important to keep this data synchronized across all identical portal clusters. See Database considerations for more information.

Each cluster can be deployed within the same data center, to help with improving maintainability and improve failure isolation, or across multiple data centers, to protect against natural disaster and data center failure, or to simply provide a broader geographical coverage of your portal site. The farther apart the clusters are, the higher the impact network latency may have between clusters and thus the less likely you will be to want to share the same physical database between clusters for the shared domains and will want to resort to database replication techniques to keep the databases synchronized.

Typically, in a multiple portal cluster topology, HTTP Servers are dedicated per cluster, since the HTTP Server plug-in's configuration is cell-specific. To route traffic between data centers (banks of HTTP Servers), a separate network load-balancing appliance is used, with rules in place to route users to specific datacenters, either based on locality or on random site selection, such as through DNS resolution. Domain, or locality, based data center selection is preferred because is predictably keeps the same user routed to the same datacenter, which helps preserve session affinity and optimum efficiency. Be mindful of the fact that DNS resolution based routing selection can cause random behavior in terms of suddenly routing users to another datacenter during a live session. If this happens, the user's experience with the portal site may be disrupted as the user is authenticated and attempts to resume at the last point in the new site. Session replication and/or proper use of portlet render parameters can help diminish this effect.

You can decide to adopt one of the following configurations with your multiple portal clusters:

- **Active/active**
  - In this configuration, all portal clusters are receiving end-user traffic simultaneously. Network load balancers and HTTP Servers help keep the traffic balanced evenly across each server in each cluster. If maintenance on one cluster is required, all production traffic is switched to the other cluster. That means that the topology must be sized such that the remaining active clusters must be able to handle all production traffic while on cluster is undergoing maintenance, or that maintenance must be performed during off-peak hours.

- **Active/passive**
  - In this configuration, under normal circumstances, all production traffic is routed to a subset of the available portal clusters (e.g. 1 of 2, or 2 of 3). There is always one cluster not receiving any traffic. Maintenance is typically applied first to the offline cluster, and then it is brought into production traffic while each of the remaining clusters are taken out and maintained in a similar fashion. This requires that a subset of the clusters be sized to handle all production traffic.
As an alternative to deploying multiple portal clusters where each cluster is in a different cell, it is also possible to deploy multiple portal clusters in the same cell. Different cells give you total isolation between clusters, and the freedom to maintain all aspects of each cluster without affecting the other. Different cells, however, require different Deployment Managers and thus different Administration Consoles for managing each cluster. Multiple clusters in the same cell reduces the administration efforts to a single console, but raises the effort level to maintain the clusters since there is a high degree of resource sharing between the multiple clusters.

While multiple portal clusters in a single cell has its uses, especially in consolidating content authoring and rendering servers for a single tier, it does increase the administrative complexity significantly. It is recommended that multiple portal clusters be deployed in multiple cells, to keep administration as simple as possible. See Multiple portal clusters in a single cell for more details.
Related concepts
Database considerations

Related tasks
Installing multiple clusters in a single cell on AIX
Installing multiple clusters in a single cell on HP-UX
Installing multiple clusters in a single cell on IBM i5/OS
Installing multiple clusters in a single cell on Linux
Installing multiple clusters in a single cell on Solaris
Installing multiple clusters in a single cell on Windows
Single-server topology for Web Content Management

Organizations that only need a small Web site or intranet site could implement a single server topology for Web Content Management. In a single server configuration, authoring and delivery occur on the same server. This configuration is not recommended for organizations that need a larger Web site or intranet site.

The following topology diagram illustrates a single-server configuration for Web Content Management. A Web server routes incoming requests from browser clients. Web Content Management, WebSphere Portal, and WebSphere Application Server are installed on the same server. The LDAP server that stores authorized user information is installed on a dedicated server. The database that stores content is also installed an a dedicated server. Although this diagram does not illustrate a cluster configuration, the Web Content Management server could be clustered. Similarly the LDAP and database servers could be clustered for failover.

The following actions occur on the single server:
- Create drafts
- Approve drafts
- Test changes
- Publish changes
- Host content

Hardware and resource considerations
- Authoring and delivering content within the same environment will be resource intensive, so the type of environment you deploy will need to be robust enough to allow authoring and delivery to occur at the same time. Using clustered servers is a common solution for a single-server configuration.
- Content can be delivered using either a Web content viewer portlet, the Web content servlet or a pre-rendered site.

Parent topic: Server topologies

Related concepts
Single-server topology
Stand-alone server topology
Horizontal cluster topology
Dual-server configuration for Web Content Management
Staging-server topology for Web Content Management
Dual-server configuration for Web Content Management

Organizations that need a large Web site with heavy traffic or that have multiple users authoring content should implement a dual-server configuration. In a dual-server configuration authoring is conducted on one server, or server cluster, and content is delivery to a different server, or server cluster. Authors access and work on the authoring server while Web site users access the delivery server. This reduces the load on each server and allows you to locate your authoring environment behind a firewall.

This diagram illustrates a dual server environment for Web Content Management. Both the delivery and authoring server access the same LDAP and database server in order to access common users, user groups, and content. Using the same LDAP configuration is critical for Web Content Management. Web site users access the site through a Web server that direct the user to the delivery server. On the delivery server you can use either the Web Content Viewer, as illustrated in the diagram, a Web content servlet, or a pre-rendered site. New content is syndicated or published to the delivery server.

On the authoring server the following actions occur:
- Create drafts
- Approve drafts
- Test changes
- Publish changes
- Syndicate changes to the delivery server

On the delivery server the following actions occur:
- Host content
- Subscribe to content changes
Hardware and resource considerations
- An authoring environment would normally be a clustered environment to cope with a large number of Web site creators and Web content authors.
- The delivery environment can be a:
  - Web content delivery server or cluster that subscribes to the authoring environment. Content can be delivered using either a Web content viewer portlet, the Web content servlet or a pre-rendered site.
  - Local WebSphere Portal production environment that subscribes to the authoring environment and delivers Web content using a Web content viewer.
  - Remote WebSphere Portal production environment that uses a remote Web content viewer to display content from the authoring environment.
  - A firewall is not depicted in this illustration, but could exist between the authoring and delivery server if desired.

Syndication options
There are two syndication options to consider:
- **Automatic one-way syndication**
  - With this option, changes made in the authoring environment are automatically syndicated to the delivery environment. You would implement this option when you want changes to your content to constantly be updated on the live site.
- **Manual one-way syndication**
  - With this option, changes made in the authoring environment are syndicated to the delivery environment in a single batch. You would implement this option when you only want to update your live site on set dates. The changes you make on your authoring server gradually accumulate until you are ready to go live with those changes.

Parent topic: Server topologies

Related concepts
- Single-server topology
- Stand-alone server topology
- Horizontal cluster topology
- Single-server topology for Web Content Management
- Staging-server topology for Web Content Management
- Design servers
- Web content authoring environments
- Web content delivery environments
Staging-server topology for Web Content Management

If you need to deliver large complex Web sites with a large number of site users and content creator, implement a staging server configuration. Implementing a staging server provides an environment for checking for accuracy, issues with the design, and performance. With a staging server configuration authoring, staging and delivery are separated onto different servers. The staging environment can be used for user acceptance testing (UAT) or to allow changes from the authoring environment to accumulate prior to syndicating the changes to the delivery environment in a single batch.

The following diagram illustrates a staging server topology. There is a different server, or cluster of servers, for the delivery, staging, and authoring environments. The delivery, staging, authoring environment all access the same LDAP and database servers. If needed the LDAP and database servers can be clustered too. Web site users access the delivery server through the Web server. Authors access the authoring server and content is syndicated to the staging server for quality assurance verification before it is published to the delivery server.

On the authoring server the following activities occur:
- Create drafts
- Approve drafts
- Test changes
- Publish new and changed content
- Syndicate content to the staging server

On the staging server the following activities occur
- User assurance testing
- Performance testing
- Syndicate content to the delivery server

On the delivery server, the following activities occur:

- Host content for Web site users

**Hardware and resource considerations**

- An authoring environment would normally be a clustered environment to cope with a large number of Web site creators and Web content authors.

- The staging environment can consist of:
  - A Web content delivery server or cluster that subscribes to the authoring environment. This would be used when you want to allow changes from the authoring environment to accumulate prior to syndicating your changes to your delivery environment in a single batch.
  - A complete replica of the delivery environment. This type of environment would be used for system UAT to ensure that the Web site being delivered is accurate, error-free and can perform under load.

- The delivery environment can consist of:
  - A Web content delivery server or cluster that subscribes to the staging environment. Content can be delivered using either a Web content viewer portlet, the Web content servlet or a pre-rendered site.
  - A local WebSphere Portal production environment that subscribes to the staging environment and delivers Web content using a Web content viewer.
  - A remote WebSphere Portal production environment. In this scenario a Web content delivery server subscribes from the staging environment, and the remote WebSphere Portal production environment uses a remote Web content viewer to display content from the Web content delivery server.

**Syndication options**

- Automatic one-way syndication from the authoring environment to the staging environment.
- Manual one-way syndication from the staging environment to the delivery environment.

**Parent topic:** Server topologies

**Related concepts**

Single-server topology
Stand-alone server topology
Horizontal cluster topology
Single-server topology for Web Content Management
Dual-server configuration for Web Content Management
Web Content Management environments

To use a Web Content Management system, you will need to use a set of Web Content Management servers. Web content servers are deployed in a production environment. A typical Web content system can include an authoring environment, a staging or preview server and a delivery environment. Each server or cluster in your Web content system requires a separate data repository, but they would usually share the same LDAP. A Web content system can be deployed in isolation, or in parallel with a WebSphere Portal production server.

- Web content system overview
  The type of Web content system you deploy will be determined by the size of your Web content system, the type of Web site being delivered and the number of users creating and viewing your Web content.

- Web content authoring environments
  An authoring environment is used to create and manage Web content. This is the environment used by your content creators and Web site designers.

- Web content staging environments
  This environment is used by your UAT testers and can be as simple as an individual UAT server where site and content updates can be tested before being syndicated to the delivery environment, or as complex as complete replica of your delivery environment where UAT can occur to both review site and content updates, and to test the performance of your delivery environment. It can also be used to accumulate changes from your authoring environment prior to syndicating your changes to your delivery environment in a single batch.

- Web content testing environments
  This environment is used by your UAT testers and can be as simple as an individual UAT server where site and content updates can be tested before being syndicated to the delivery environment, or as complex as complete replica of your delivery environment where UAT can occur to both review site and content updates, and to test the performance of your delivery environment. It can also be used to accumulate changes from your authoring environment prior to syndicating your changes to your delivery environment in a single batch.

- Web content delivery environments
  A delivery environment is used to deliver content to your Web site viewers.

Parent topic: Planning for WebSphere Portal
Web content system overview
The type of Web content system you deploy will be determined by the size of your Web content system, the type of Web site being delivered and the number of users creating and viewing your Web content.

Web content system types
There are three main types of Web content systems:

- **Single environment systems**
  - This is where authoring and delivery occur within a single environment. This type of environment would be deployed by a small organization delivering a small Web site, such as an intranet. Authoring and delivering content within the same environment will be resource intensive, so the type of environment you deploy will need to be robust enough to allow authoring and delivery to occur at the same time. For example, using clustered servers is a common solution for a single instance system.

- **Dual environment systems**
  - This is where authoring and delivery are split into different environments. This reduces the load on both authoring and delivery and also allows you to locate your authoring environment behind a firewall. This type of system would be used when delivering externally facing Web sites, or where you have a large number of users authoring content or a large number of users viewing a Web site.

- **Staged systems**
  - This is where a staging environment is added between the authoring and delivery environments. The staging environment can be used for user acceptance testing (UAT) or to allow to accumulate changes from your authoring environment prior to syndicating your changes to your delivery environment in a single batch. This system would be deployed when delivering large, complex sites with a large number of content creators and you need to ensure that the Web site being delivered is accurate, error-free and can perform under load.
Environment types

- **Authoring environment**
  - An authoring environment is used to create and manage Web content. This is the environment used by your content creators and Web site designers. An authoring system can consist of:
    - an authoring server or cluster.
    - individual UAT servers where site and content updates can be tested before being syndicated to the delivery environment.

- **Staging environment**
  - A staging environment can consist of:
    - individual holding servers where changes from your authoring environment can be accumulated prior to syndicating your changes to your delivery environment in a single batch. Pairs of holding servers can be used to provide you with built-in redundancy.
    - a complete replica of your delivery environment where UAT can occur to both review site and content updates, and to test the performance of your delivery environment.

- **Delivery environment**
  - This environment is used by your Web site viewers. A delivery environment can consist of:
    - pre-rendered sites where a Web content site is pre-rendered as a set of HTML files which are then used to deliver a static Web site.
    - a WebSphere Portal server or cluster where content is delivered using a servlet. Servlet delivery is used to deliver Web sites that contain dynamic content, but don’t include any WebSphere Portal content or applications.
    - a WebSphere Portal server or cluster where content is delivered using either a local or remote Web Content Viewer portlet. Web Content Viewer portlets are used to deliver Web sites that contain dynamic Web content alongside other WebSphere Portal portlets or applications.
    - A combination of all of the above.

**Parent topic:** Web Content Management environments
Web content authoring environments

An authoring environment is used to create and manage Web content. This is the environment used by your content creators and Web site designers.

Server type

Most Web Content Management sites will need to support a large number of content creators. A clustered server solution is the best solution for this scenario.

Standard authoring environment

A standard authoring environment consists of a single authoring cluster that syndicates directly to either a staging or delivery environment. For example:

Authoring environment with testing

You can add a test environment to your authoring environment to enable you to perform user acceptance testing on your content management system and Web site. For example:

Decentralized authoring environments

If you content creators are located at different locations, or you have different groups of content creators, you might
consider deploying a set of decentralized authoring clusters. This scenario works best if the decentralized content creators work with separate content stored in different content libraries. For example, if you have users located in different locations it may be more efficient to install a local authoring environment at each location. Two-way syndication is used between all authoring environments with a centralized authoring environment. This allows changes made at all remote locations to be visible to all users.

Decentralized authoring creates the risk of conflicting updates between authoring environments. This may be managed by allocating different sites, or different sections of a site to each authoring environment. You could also use different authoring environments for different user roles. For example, Content creators could use a different authoring environment than presentation template designers.

Access to each decentralized authoring environment is controlled using a combination of authoring portlet access controls and item security settings. For example, Only users requiring access to the local authoring environment would be granted access to the local authoring portlet. Users would be given "Read" access to all items, but only "Edit" access to items they are required to update.

**Parent topic:** Web Content Management environments

**Related concepts**

- Dual-server configuration for Web Content Management
- Syndication
Related tasks
Setting up a cluster
Web content staging environments

This environment is used by your UAT testers and can be as simple as an individual UAT server where site and content updates can be tested before being syndicated to the delivery environment, or as complex as complete replica of your delivery environment where UAT can occur to both review site and content updates, and to test the performance of your delivery environment. It can also be used to accumulate changes from your authoring environment prior to syndicating your changes to your delivery environment in a single batch.

- **Built-in redundancy**
  
  Syndication can be used to supply built-in redundancy to a group of Web Content Management applications.

Parent topic: Web Content Management environments
**Built-in redundancy**
Syndication can be used to supply built-in redundancy to a group of Web Content Management applications.

**Example 1: Automatic redundancy**
In this example, a pair of Staging applications are used to syndicate between authoring and delivery applications. Syndication is enabled on both staging applications to each authoring and delivery application. This means that each staging application contains the same data. If one staging application goes down, data continues to be automatically syndicated from the authoring application to the delivery application via the remaining staging application.

This architecture is useful when you are continuously syndicating between authoring, staging and delivery applications. This automatic redundancy structure could also be applied to a set of authoring or delivery applications.

**Example 2: Manual redundancy**
In this example, a primary Staging application is used to syndicate to a Delivery application. This primary Staging application also syndicates to a backup application. If the primary Staging application goes down, you can manually switch to the backup Staging application by enabling syndication between the backup Staging application and any Authoring or Delivery applications.

This architecture is useful if you only manually "batch-syndicate" from a Staging application to a Delivery application. This manual redundancy structure could also be applied to a set of authoring or delivery applications.

**Note:** Syndication is used as the transport layer that replicates data from one IBM® Lotus Web Content Management Portal, Version 6.1.5 Operating systems: AIX, HP-UX, i5/OS, Linux, Solaris, Windows
application to another. Further information can be found in Syndication.

**Parent topic:** Web content staging environments
Web content testing environments
This environment is used by your UAT testers and can be as simple as an individual UAT server where site and content updates can be tested before being syndicated to the delivery environment, or as complex as complete replica of your delivery environment where UAT can occur to both review site and content updates, and to test the performance of your delivery environment. It can also be used to accumulate changes from your authoring environment prior to syndicating your changes to your delivery environment in a single batch.

Site testing within an authoring environment
You can add a testing environment to your authoring environment to enable user acceptance testing to be performed on your content management system and your Web site:

System testing within a staging environment
You can perform system testing on a replica of your delivery environment prior to syndicating your changes to the live delivery environment:

Parent topic: Web Content Management environments
Web content delivery environments
A delivery environment is used to deliver content to your Web site viewers.

Pre-rendered delivery
You can pre-render a complete Web site into HTML and save it to disk. The pre-rendered site can then be used as your live site and displayed to end users using either Web Content Management or a Web server. You deploy a pre-rendered site when you are not using any WebSphere Portal features, such as portlets, and your content is static and is only updated periodically.

Servlet delivery
Users can access content displayed via the Web Content Management servlet. A servlet delivered Web site should be used when you don't need to use any WebSphere Portal based features such as authoring tools.

Local Web content viewer delivery
Web content viewers are portlets that display content from a Web content library as part of a portal page. If your presentation is simple, a single Web content viewer can be sufficient, but you can also use multiple Web content viewers to aggregate content from different libraries and provide a richer experience for your users. A local Web content view portlet is used to display content within your Web content delivery environment.
Remote Web content viewer delivery

A remote Web content view portlet is used to display content on a remote WebSphere Portal server or cluster.

Parent topic: Web Content Management environments

Related concepts
Dual-server configuration for Web Content Management
Web servers

By default IBM® WebSphere® Portal uses the internal HTTP transport within IBM WebSphere Application Server to handle requests. However, because WebSphere Application Server also supports the use of an external Web server, you can access WebSphere Portal from your Web server. You can use a local Web server on the same machine as WebSphere Portal or you can use a remote Web server on a different machine. A remote Web server is typical for a production environment or other high-traffic configuration and is also typically placed in demilitarized zones (DMZ) outside a firewall to protect portal ports.

To enable communication between the Web server and WebSphere Application Server, a Web server plug-in is required. The Web server plug-in determines whether a request is handled by the Web server or by the application server. The plug-in can be installed into a Web server that is located either on the same machine as WebSphere Application Server or on a separate machine. The Web server plug-in uses an XML configuration file (plugin-cfg.xml) that contains settings that describe how to handle and pass on requests to the WebSphere Application Server made accessible through the plug-in. In the WebSphere Application Server administrative console, the Web server is represented as a specific server type, and you can view or modify all of the configuration properties used in the plugin-cfg.xml file for the Web server plug-in from the administrative console.

**Note:** For some portal functions to work you need to make sure that write and delete operations are permitted by the Web server so that the HTTP operations POST, PUT, and DELETE are enabled. For example, this is required for mashup integration.

**i5/OS users:** For detailed information on using an external Web server with your i5/OS system, see Selecting a Web server topology diagram and roadmap, in addition to the steps listed on this page.

**Access WebSphere Portal through another HTTP port in a stand-alone or clustered environment**

By default WebSphere Portal is configured to be accessed through the internal HTTP port in WebSphere Application Server. For example, http://hostname.example.com:10040/wps/portal, where hostname.example.com is the fully qualified host name of the machine where WebSphere Portal is running and 10040 is the default transport port that is created by WebSphere Application Server; the port number may be different for your environment. The default host name and port used by WebSphere Portal are specified by the WpsHostName and WpsHostPort properties in the wkplc.properties file.

After configuring WebSphere Portal to use an external Web server, you will access the portal with the Web server host name and port (for example, 80). For stand-alone servers or vertical cluster members, you will be unable to access the portal using the WebSphere Portal host name and port (for example, 10040), unless there is a corresponding virtual host definition for port 10040 in the WebSphere Application Server configuration.

Many of the WebSphere Portal configuration tasks rely on the WpsHostName and WpsHostPort properties from the wkplc.properties file. You must ensure that WebSphere Portal can be accessed using the host name and port specified by these property values. You can do this in one of two ways:

- Modify the WpsHostName and WpsHostPort property values to specify the Web server host name and port.
- Add the appropriate virtual host definition, as described below.

If you want to access WebSphere Portal using a host name and port different from your Web server, add the required virtual host definition using the WebSphere Application Server administrative console. If you are using the Web server in a clustered environment, use the deployment manager administrative console to perform these steps.
1. Select **Environment > Virtual Hosts**.

2. Select the **default_host** entry or the entry for the virtual host that is being used to access the WebSphere Portal application.

3. Select **Host Aliases**, and verify whether there is a host name and port entry corresponding to the values used to access WebSphere Portal (for example, `*:10040`). If the entry does not exist, select **New**, and enter the information for the host name and port you want to use.

4. Save your changes.

5. Regenerate the Web server plug-in.

6. If you are using a remote Web server, copy the updated `plugin-cfg.xml` file to the Web server machine.

7. If you are running a system under stress and are expecting requests to take longer than the **ServerIOTimeout** default value, you should increase this value to avoid requests being sent twice.

8. Recycle your Web server, and your portal.

9. In a clustered environment, resynchronize the nodes and restart the cluster.

### Cluster considerations for Web servers

When using a Web server in a clustered environment with WebSphere Portal, the following considerations apply:

- If you run the `configure_web_server_name` script on the deployment manager system, you must synchronize and restart the cluster to ensure proper communication between the Web server and the cluster members.

- If a standalone WebSphere Portal node was previously configured for a web server and then federated into a cell managed by the deployment manager, the web server definitions on that node will be removed. If you want to use the Web server with the cluster, you must recreate the Web server definition in the deployment manager administration console after you federate the node.

**Parent topic:** Planning for WebSphere Portal
Database considerations


Data storage and failover become more complex as the network environment becomes complex. In a complex and high-demand environment, data may be distributed across multiple database servers for high-volume storage and failover configuration. For example, in a production environment, the demand for quick response time in a high-demand situation combined with the desire for failover capability distributed across multiple database servers is likely to require transferring the database to a more robust server. Therefore, learn the limitations of using Derby and determine how transferring to another database affects the capacity and scalability of an environment. Also consider that the version of Derby provided with WebSphere Portal has stress and failover limitations that are expected to be resolved in a later release of Derby performance improvements.

Derby is a built-in Java database that provides a small footprint, is self-tuning, and is ideal for solutions where the database must be hidden. Derby works in a non-clustered environment with a small number of users, such as a portlet development or testing environment or a proof-of-concept environment.

The Derby database that is installed by default is not intended for use in a production environment or for authoring Web content. While using one database is supported, for high performance and availability, use multiple databases. Derby does not support clustered environments, enabling security in a database-only mode, or vertical cloned environments in which multiple application servers are configured on a single server. Use one of the other supported databases in a production environment or when authoring Web content because they are better able to handle large amounts of data and can be tuned for performance.

With appropriate tuning, other databases can provide performance gains. Other databases support vertical and horizontal clusters and cloning. Use multiple databases for high performance and availability.

Other databases require increased CPU and memory resources and add complexity to the configuration environment. When you choose to transfer data to another supported database, perform the database transfer before you use the portal extensively. Large amounts of data in the databases can cause the database transfer to fail if your Java heap size is not large enough. Because information is added to the databases as you use the portal, perform database transfer as soon as you realize that a high volume of data needs to be stored and failover is necessary. Transferring the database sooner rather than later avoids the problems typically caused by transferring a high volume of data at a later stage to the other supported databases, including not having adequate Java heap size.

Data can be transferred from a Derby database, but cannot be transferred to a Derby database. If you are transferring from a database other than the default database, you will need to edit the wkplc_comp.properties and wkplc_dbtype.properties files to update the source and target database information.

- **Database topologies**

  Consider the database configuration options in relation to your IBM WebSphere Portal deployment scenario. The complexity of the network topology will increase as you scale from a proof-of-concept environment using Derby to systems using vertical and horizontal cloning techniques.

- **Shared database domains**

  Database domains classify and help you determine how to distribute portal data. To maximize data availability, you can distribute portal data across multiple databases and, for some domains, share data between multiple lines of production. You can choose to transfer a single database domain or multiple domains.
- **Database compatibility considerations**
  If you are a database administrator, consider the database requirements for IBM WebSphere Portal.

- **Planning for DB2 on Windows**
  When planning to transfer data to IBM DB2 Universal Database Enterprise Server Edition, you should consider the databases and user information, such as database names, what data is stored, and the database space needed. Some fix packs require steps prior to the transfer task to complete successfully.

- **Planning for DB2 on UNIX or Linux**
  When planning to transfer data to IBM DB2 Universal Database Enterprise Server Edition, you should consider the databases and user information, such as database names, what data is stored, and the database space needed. Some fix packs require steps prior to the transfer task to complete successfully.

- **Planning for DB2 for i5/OS**
  When planning to transfer data to DB2 for i5/OS, you should consider the databases and user information, such as database names, what data is stored, and the database space needed. Some fix packs require steps prior to the transfer task to complete successfully.

- **Planning for DB2 for z/OS**
  When planning to transfer data to IBM DB2 Universal Database for z/OS®, you should consider the databases and user information, such as database names, what data is stored, and the database space needed. Some fix packs require steps prior to the transfer task to complete successfully.

- **Planning for Oracle**
  When planning to transfer data to Oracle, you should consider the databases and user information, such as database names, what data is stored, and the database space needed.

- **Planning for Oracle RAC**
  When planning to transfer data to Oracle RAC, you should consider the databases and user information, such as database names, what data is stored, and the database space needed.

- **Planning for SQL Server 2005**
  When planning to transfer data to SQL Server, you should consider the databases and user information, such as database names, what data is stored, and the database space needed.

**Parent topic:** Planning for WebSphere Portal
Database topologies

Consider the database configuration options in relation to your IBM® WebSphere® Portal deployment scenario. The complexity of the network topology will increase as you scale from a proof-of-concept environment using Derby to systems using vertical and horizontal cloning techniques. WebSphere Portal data can be configured in a single store or organized into database domains to meet different availability requirements, depending on your deployment scenarios for proof-of-concept and production environments. The database domains provided by WebSphere Portal help you classify and distribute portal data. Each database domain can be placed on a separate database for efficient maintenance. Additionally, each domain can be placed on a separate database server system for maximum performance.

Consider whether your portal deployment requires a local database for a proof-of-concept environment; remote databases that share database domains to support normal load balancing; remote databases for high-capacity, heavy load balancing that separate data; or a combination of database configurations.

- **Proof-of-concept deployment using a local database**
  - You can install the database server on the same server machine as WebSphere Portal. This is typically referred to as a local database. Using a local database can make administering your environment easier; however, this setup is best used for proof-of-concept deployments only.

- **Normal load balancing using one or more remote databases**
  - You can install the database server on a different server machine from WebSphere Portal. This is typically referred to as a remote database. Using a remote database can provide performance benefits, depending on the speed of the network. When multiple lines of production are involved and each line of production is implemented as a cluster of servers, sharing database domains ensures that the data is automatically synchronized across the lines of production. Each database domain can be placed on a separate database for efficient maintenance.

- **High-capacity load balancing using one or more remote databases**
  - When you deploy the portal in a large-scale, high-demand environment, you can dedicate a server specifically for database transactions. As more users access the portal, the portal application becomes database intensive. Database activity can take up CPU utilization and disk I/O time. Separating the database from the server that the portal is running on increases its capacity. Each domain can be placed on a separate database server system for maximum performance.

*Remember:* If you install the database server on a remote system, you need to install the database client software on the
WebSphere Portal system so that the portal can communicate with the remote database server.

**Parent topic:** Database considerations
Shared database domains

Database domains classify and help you determine how to distribute portal data. To maximize data availability, you can distribute portal data across multiple databases and, for some domains, share data between multiple lines of production.

You can choose to transfer a single database domain or multiple domains.

Separation of data allows you to store each category of data in its own set of database tables or the file system. The sets of databases tables and schemas for portal resources are called database domains. Database domains support the storage and transfer of data by category, for example, Configuration, Release, Customization, Community, and IBM Java Content Repository (JCR). Separating your data allows you to share domains across multiple portals. You can also spread the different domains across different database types. For example, you can choose to leave LikeMinds data on your default database and move all other data to another database. The separation of the domains can be used to support production environments, where the production nodes are split into separate clusters. Each cluster can run independently, but share the same Community and Customization database domains, for example. Each of these clusters is called a line of production.

Preferences are kept in layers that are modifiable based on portlet modes. For example, there is one layer of default preferences defined by the portlet deployment descriptor. This layer is modifiable within the CONFIG mode supported by WebSphere Portal. In WebSphere Application Server, the values of the portlet deployment descriptor are read-only.

WebSphere Portal provides one additional preference layer that enables portal administrators to specify different default values per portlet window. This capability is supported through the portlet mode EDIT_DEFAULTS, and applies to all who use the same portlet window. There is no such preference layer in WebSphere Application Server. Both products support the standard modes: VIEW, EDIT, and HELP. When a user customizes a portlet on a page in any standard mode, the user can change his personal portlet preferences. Default preferences on a per page or per portlet base cannot be set in any standard mode; you need to use custom portlet modes instead. Portlet preferences are stored in the customization domain when stored by users (typically in edit mode) on the entity level whereas when using configure mode, we're working on the portlet definition level and those are stored on the release level.

The database domains categorize portal data into the following categories and subcategories to help you decide how to distribute portal data into different databases:

- **Configuration data**
  - This data defines the portal server setup, such as database connection, object factories, and deployment descriptors. This type of data typically is constant during the time a server node is running. Configuration data is typically kept in property files and is either protected by file system security or application server administration rights.

- **Release data**
  - This data includes all portal content definitions, rules, and rights that are designed externally then brought into the portal by a staging process, such as Page Hierarchy, available Portlets and Themes, Templates, Credential Slots, Personalization Rules, and Policies. These resources are typically not modified during production and need administrative rights to do so. Administrators typically create release data on an integration server and stage it to the production system. Release data are protected by access control and contain only data, not code. In an environment that consists of multiple lines of production, one copy of the release data exists per cluster. Release data includes the following two separate domains:
    - Release: Contains portal's static site configuration, including access control, pages, and portlets.
    - JCR: Contains authored content, Personalization rules, and theme policy definitions.

    These should be considered release data and promoted to production lines individually.
- **Customization data**
  - This data is associated with a particular user only and cannot be shared across users or user groups. Typical examples are Portlet Data or Customized Pages (Implicitly Derived Pages). Because this data is scoped to a single user only, access control protection is simplified.
  - In an environment that consists of multiple lines of production, customization data is kept in a database that is shared across the lines of production. Therefore the data is automatically in synchronization across the lines of production.

- **Community data**
  - This data includes all resources that are usually modified during production, such as Composite Application instances. Typically, users and groups are allowed to modify or delete shared resources. Community resources are protected by access control.

The following table lists the supported database domains, whether a domain is sharable, and its storage method. *Table 1. Supported database domains*

<table>
<thead>
<tr>
<th>Database domain</th>
<th>Sharable</th>
<th>Storage method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release</td>
<td>no</td>
<td>database</td>
</tr>
<tr>
<td>Customization</td>
<td>yes</td>
<td>database</td>
</tr>
<tr>
<td>Community</td>
<td>yes</td>
<td>database</td>
</tr>
<tr>
<td>JCR</td>
<td>no</td>
<td>database</td>
</tr>
<tr>
<td>Feedback</td>
<td>yes</td>
<td>database</td>
</tr>
<tr>
<td>LikeMinds</td>
<td>yes</td>
<td>database</td>
</tr>
</tbody>
</table>

For maintenance and staging purposes you can take a single line of production out of service while another line is still serving requests with the old data. After the first production line is updated and back in service again, the second line is updated using the same approach. Updates of data in the shared domain are critical because they influence the other production line.

The capacity of the entire environment should be greater than the intended use so that individual servers can be taken out of production without affecting application availability. To ensure that all of the system resources are available for the portal, production systems should be dedicated to the portal and should not run any other server software that is not related to the portal.

For maintenance purposes, the following database domains can be taken offline:
- Community
- Customization
- Feedback
- LikeMinds

The following databases must not be taken offline at anytime when WebSphere Portal is started:
- Release
- JCR

While a database domain is offline, WebSphere Portal cannot access the corresponding data and thus error messages may be displayed. WebSphere Portal itself remains responsive. When a database domain becomes available again, WebSphere Portal will detect this availability, reconnect, and continue working with the corresponding data. Regular maintenance should not affect the shared database domains because it is imperative that this data remain available to all lines of production currently in use.

**Sharing of VMM databases**

The VMM database feature makes it much simpler to use multiple repositories, since this capability is achieved through configuration, rather than development, with the use of the new VMM. In essence, this feature provides the ability to map entries from multiple individual user repositories into a single virtual repository. The federated repository consists of a
single named realm, which is a set of independent user repositories. Each repository may be an entire external repository or, in the case of LDAP, a subtree within that repository. The root of each repository is mapped to a base entry within the federated repository, which is a starting point within the hierarchical namespace of the virtual realm. The Virtual Member Manager (VMM) databases for a full repository and for the property extension can be shared between lines of production. If the VMM databases are out of service, WebSphere Portal will not function.

**Parent topic:** Database considerations
Database compatibility considerations

If you are a database administrator, consider the database requirements for IBM® WebSphere® Portal. Remember that database compatibility with IBM WebSphere Portal running on the IBM i5/OS operating system is mutually exclusive:

- IBM DB2 Universal Database™ for iSeries® databases are supported only on WebSphere Portal for i5/OS.
- Other database management systems that are supported by WebSphere Portal can be configured to work with any portal platform.

Parent topic: Database considerations

Related tasks
Deleting passwords from properties files
Planning for DB2 on Windows

When planning to transfer data to IBM® DB2 Universal Database™ Enterprise Server Edition, you should consider the databases and user information, such as database names, what data is stored, and the database space needed. Some fix packs require steps prior to the transfer task to complete successfully.

Before you begin:
- WebSphere Portal supports DB2 JDBC Type 2 (CLI-based) and Type 4 (JCC) drivers.
- Review the database considerations.
- Ensure the database that you plan to use is supported by this version of WebSphere Portal. Refer to the list of supported databases in the WebSphere Portal detailed system requirements.

The database names and users on this page are suggested values and provide consistency throughout the documentation. Replace these values with values in your environment. The database name cannot exceed eight characters and can only contain letters and numbers.
- When using the same database user ID, the value for the database name, database server node, or schema name must be unique.
- When the DB2 Universal JDBC driver (type 4 mode) is used, connect to the database directly. Do not connect to an alias database (gateway), instead specify the real database name in the JDBC connection URL (`dbdomain.DbUrl`) and in the database name property (`dbdomain.DbName`).

In a local database environment, WebSphere Portal and DB2 are installed on the same system.

![Figure 1. Local database environment](image1)

As shown in Figure 2, when a JDBC type 2 connections is used, WebSphere Portal and DB2 Connect are installed on one system (the local system). The DB2 server is installed on a separate system (the remote system).

![Figure 2. Remote Database Environment (JDBC type 2 connection)](image2)
For JDBC type 4 connections, no DB2 Connect installation is required on the system that runs WebSphere Portal. As shown in Figure 3, the DB2 Universal JDBC driver that is supplied with DB2 is copied to this system. It is used within the Java Virtual Machine of WebSphere Portal and connects directly to the remote DB2 server.

**Figure 3. Remote Database Environment (JDBC type 4 connection)**

1. Review the different databases shown in the following table and replace these values with the values in your environment; schema names must be different when the database is shared. While configuring WebSphere Portal to use one database is technically possible, using separate databases can improve scalability and performance. The architecture allows each of these databases to exist in one or many instances. However, the recommended architecture uses the default instance (db2inst1) that is created by the installation program. *Table 1. Space required for various databases*

<table>
<thead>
<tr>
<th>Application</th>
<th>Database name</th>
<th>Space required</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Portal</td>
<td>relbcommdbcustdb</td>
<td>Depends on the number of users and portal objects, such as pages and portlets.</td>
</tr>
<tr>
<td>Personalization, Web Content Management</td>
<td>jcrdb</td>
<td>Depends on the number and size of Personalization rules and campaigns, and the number and size of items and elements created in Web Content Management.</td>
</tr>
<tr>
<td>Feedback</td>
<td>fdbkdb</td>
<td>Depends on the amount of traffic to the site. The amount of data that is logged per login-enabled page can vary.</td>
</tr>
<tr>
<td>LikeMinds</td>
<td>lmdb</td>
<td>Depends on the amount of traffic to the site.</td>
</tr>
</tbody>
</table>

2. Review the tables and types of objects owned by each user. The architecture allows each of the following users to exist in the same database. All table spaces are approximately 2.8 GB by default. The size increases with the use of Java Content Repository. *Table 2. Tables and objects owned by database users*
<table>
<thead>
<tr>
<th>WebSphere Portal</th>
<th>releaseusrcommunityusrcustomizationusr</th>
<th>Core user who owns approximately 230 tables, used for WebSphere Portal core objects, which includes tables that store the user customizations made to pages.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java Content Repository</td>
<td>jcr</td>
<td>Java Content Repository user who owns at least 1130 tables. The number could be higher depending on usage.</td>
</tr>
<tr>
<td>Feedback</td>
<td>feedback</td>
<td>Feedback user who owns approximately 50 tables used for logging site and personalization usage.</td>
</tr>
<tr>
<td>LikeMinds</td>
<td>likeminds</td>
<td>LikeMinds user who owns approximately 15 tables used to hold the Web site usage analysis routines and recommendation text.</td>
</tr>
</tbody>
</table>

**Parent topic:** Database considerations

**Related concepts**

Supported hardware and software

**Related Information**

- Which distributed edition of DB2 Universal Database Version 8 is right for you?
- Which distributed edition of DB2 Universal Database Version 9 is right for you?
Planning for DB2 on UNIX or Linux

When planning to transfer data to IBM® DB2 Universal Database™ Enterprise Server Edition, you should consider the databases and user information, such as database names, what data is stored, and the database space needed. Some fix packs require steps prior to the transfer task to complete successfully.

Before you begin:
- WebSphere Portal supports DB2 JDBC Type 2 (CLI-based) and Type 4 (JCC) drivers.
- Review the database considerations.
- Ensure the database that you plan to use is supported by this version of WebSphere Portal. Refer to the list of supported databases in the WebSphere Portal detailed system requirements.

The database names and users on this page are suggested values and provide consistency throughout the documentation. Replace these values with values in your environment. The database name cannot exceed eight characters and can only contain letters and numbers.
- When using the same database user ID, the value for the database name, database server node, or schema name must be unique.
- Accessing a local DB2 database can cause shared memory problems. To correct these problems, you must treat the local database as a remote database on the local system. Follow the instructions in the Creating remote databases section if you are manually creating a database.
- When the DB2 Universal JDBC driver (type 4 mode) is used, connect to the database directly. Do not connect to an alias database (gateway), instead specify the real database name in the JDBC connection URL (\texttt{dbname valide DbUrl}) and in the database name property (\texttt{dbname valide DbName}).

In a local database environment, WebSphere Portal and DB2 are installed on the same system.

As shown in Figure 2, when a JDBC type 2 connections is used, WebSphere Portal and DB2 Connect are installed on one system (the local system). The DB2 server is installed on a separate system (the remote system).

![Figure 1. Local database environment](image)

![Figure 2. Remote Database Environment (JDBC type 2 connection)](image)
For JDBC type 4 connections, no DB2 Connect installation is required on the system that runs WebSphere Portal. As shown in Figure 3, the DB2 Universal JDBC driver that is supplied with DB2 is copied to this system. It is used within the Java Virtual Machine of WebSphere Portal and connects directly to the remote DB2 server.

**Figure 3. Remote Database Environment (JDBC type 4 connection)**

1. Review the different databases shown in the following table and replace these values with the values in your environment; schema names must be different when the database is shared. While configuring WebSphere Portal to use one database is technically possible, using separate databases can improve scalability and performance. The architecture allows each of these databases to exist in one or many instances. However, the recommended architecture uses the default instance (db2inst1) that is created by the installation program. *Table 1. Space required for various databases*

<table>
<thead>
<tr>
<th>Application</th>
<th>Database name</th>
<th>Space required</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Portal Used for the portal (at a minimum) or to hold all data. Stores information about user customization, such as pages, and user profile and login information.</td>
<td>relbcommdbcustdb</td>
<td>Depends on the number of users and portal objects, such as pages and portlets.</td>
</tr>
<tr>
<td>Personalization, Web Content Management Contains documents, personalization rules, personalization campaigns, and document library configuration information.</td>
<td>jcrdb</td>
<td>Depends on the number and size of Personalization rules and campaigns, and the number and size of items and elements created in Web Content Management.</td>
</tr>
<tr>
<td>Feedback Contains the information that is logged by your Web site for analysis of site activity and generating reports.</td>
<td>fdbkdb</td>
<td>Depends on the amount of traffic to the site. The amount of data that is logged per login-enabled page can vary.</td>
</tr>
<tr>
<td>LikeMinds Contains the recommendations to be displayed to users when their interactions with your Web site have been analyzed and predictions generated.</td>
<td>lmdb</td>
<td>Depends on the amount of traffic to the site.</td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td><strong>releaseusrcommunityusrcustomizationusr</strong></td>
<td>Core user who owns approximately 230 tables, used for WebSphere Portal core objects, which includes tables that store the user customizations made to pages.</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Java Content Repository</strong></td>
<td><strong>jcr</strong></td>
<td>Java Content Repository user who owns at least 1130 tables. The number could be higher depending on usage.</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td><strong>feedback</strong></td>
<td>Feedback user who owns approximately 50 tables used for logging site and personalization usage.</td>
</tr>
<tr>
<td><strong>LikeMinds</strong></td>
<td><strong>likeminds</strong></td>
<td>LikeMinds user who owns approximately 15 tables used to hold the Web site usage analysis routines and recommendation text.</td>
</tr>
</tbody>
</table>

**Parent topic:** Database considerations

**Related concepts**

Supported hardware and software

**Related Information**

- Which distributed edition of DB2 Universal Database Version 8 is right for you?
- Which distributed edition of DB2 Universal Database Version 9 is right for you?
Planning for DB2 for i5/OS

When planning to transfer data to DB2 for i5/OS, you should consider the databases and user information, such as database names, what data is stored, and the database space needed. Some fix packs require steps prior to the transfer task to complete successfully.

Before you begin:
- Review the database considerations.
- Use the IBM i command, WRKRDBDIRE, to confirm there is a database entry for a local database environment or for a remote database in remote environment since the default database connection is a type 4 connection.
- Ensure the database that you plan to use is supported by this version of WebSphere Portal. Refer to the list of supported databases in the WebSphere Portal detailed system requirements.

DB2 is integrated with i5/OS, but you must create the required databases and users and grant the proper privileges to those users. WebSphere Portal can create the databases for you.

The database names and users in this topic are suggested values and provide consistency throughout the documentation. Replace these values with values in your environment.

WebSphere Portal uses IBM® Derby during installation, and supports configuration of DB2 for an IBM System i5 system. The Derby database is ideal for a test environment. For a production environment, you can move from the installed Derby database to DB2 for i5/OS, but note that there is no option to transfer back to Derby.

The DB2 for i5/OS database enables you to access and manage server data through an application or a user interface. In addition to providing access to and protection for your data, IBM DB2 Universal Database™ for iSeries® provides advanced functions, such as referential integrity and parallel database processing.

DB2 for i5/OS is the relational database manager that is fully integrated on your i5/OS system. DB2 for i5/OS also provides features such as triggers, stored procedures, and dynamic bitmapped indexing that serve a wide variety of application types. These applications range from traditional host-based applications to client/server solutions to business intelligence applications.

As an interface to DB2 for i5/OS, the DB2 Query Manager and Structured Query Language (SQL) Development Kit for System i5 add an interactive query and report writing interface, and precompilers and tools to assist in writing SQL application programs in high-level programming languages. The SQL implementation for OS/400 allows you to define, manipulate, query, and control access to your i5/OS data. It works equally well with i5/OS files and SQL tables.

If you choose to use one database to hold all WebSphere Portal, Member Manager, and content publishing information, only one user profile is required. Additional user profiles are necessary only if using multiple i5/OS systems or separate databases are required.

When WebSphere Portal creates databases, it uses the database names that are specified in the wkplc_comp.properties file, located in the UserData directory wp_profile_root/ConfigEngine/properties. It is possible to create up to six different databases by setting different values in the wkplc_comp.properties file.

While configuring Portal to use one database is technically possible, use separate databases for scalability and performance tuning reasons. To use a single shared database, replace each database and user variable with the name of your database and database user, respectively. Note: The format for database names is "LOCAL/database_name" for a Type 2 database connection, or a fully qualified server name such as myserver.mycompany.com/database_name for a Type 4 database connection.

In a local database environment, WebSphere Portal and DB2 for i5/OS can be accessed with either a Type 4 or Type 2...
connection. Type 4 is the default and recommended connection.

In a remote database environment, WebSphere Portal connects directly to the DB2 server (JDBC Type 4 connection).

1. Review the different databases shown in the following table and replace these values with the values in your environment; schema names must be different when the database is shared. Table 1. Space required for various databases

<table>
<thead>
<tr>
<th>Application</th>
<th>Database name</th>
<th>Space required</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Portal</td>
<td>reldbcocommmdbcustodb</td>
<td>Depends on the number of users and portal objects, such as pages and portlets.</td>
</tr>
<tr>
<td>Personalization, Web Content Management</td>
<td>jcrdb</td>
<td>Depends on the number and size of Personalization rules and campaigns, and the number and size of items and elements created in Web Content Management.</td>
</tr>
<tr>
<td>Feedback</td>
<td>fdbkdb</td>
<td>Depends on the amount of traffic to the site. The amount of data that is logged per login-enabled page can vary.</td>
</tr>
<tr>
<td>LikeMinds</td>
<td>lmdb</td>
<td>Depends on the amount of traffic to the site.</td>
</tr>
</tbody>
</table>

2. Review the tables and types of objects owned by each user. The architecture allows each of the following users to exist in the same database. All table spaces are approximately 2.8 GB by default. The size increases with the use of Java Content Repository. Table 2. Tables and objects owned by database users

<table>
<thead>
<tr>
<th>Application</th>
<th>Database user placeholder</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Portal</td>
<td>releaseusrcommunityusrcustomizationusr</td>
<td>Core user who owns approximately 230 tables, used for WebSphere Portal core objects, which includes tables that store the user customizations made to pages.</td>
</tr>
<tr>
<td>Java Content Repository</td>
<td>jcr</td>
<td>Java Content Repository user who owns at least 1130 tables. The number could be higher depending on usage.</td>
</tr>
<tr>
<td>Feedback</td>
<td>feedback</td>
<td>Feedback user who owns approximately 50 tables used for logging site and personalization usage.</td>
</tr>
<tr>
<td>LikeMinds</td>
<td>likeminds</td>
<td>LikeMinds user who owns approximately 15 tables used to hold the Web site usage analysis routines and recommendation text.</td>
</tr>
</tbody>
</table>
Parent topic: Database considerations

Related information

http://www.ibm.com/support/docview.wss?rs=688&uid=swg27007791

| | |
Planning for DB2 for z/OS

When planning to transfer data to IBM® DB2 Universal Database™ for z/OS®, you should consider the databases and user information, such as database names, what data is stored, and the database space needed. Some fix packs require steps prior to the transfer task to complete successfully.

When planning to install DB2 for z/OS to use as the database for WebSphere Portal, consider the following:

- Review the database considerations.
- Ensure the database that you plan to use is supported by this version of WebSphere Portal. Refer to the list of supported databases in the WebSphere Portal detailed system requirements.
- Ensure that Java Database Connectivity requirements are met. Consult the following references:
  - DB2 Universal Database for OS/390 and z/OS: Application Programming Guide and Reference for Java
  - The IBM Redbook, DB2 for z/OS and OS/390: Ready for Java SG24-6435-00.
- If you plan to use IBM DB2 Universal Database for z/OS for a database transfer, as a database user registry, or a property extension (looksaside) database, change the Common Service Area (CSA) setting to 3500,350000. See the appropriate DB2 for z/OS Information Center topic for complete information about calculating and setting CSA:
  - DB2 for z/OS Version 8, Common service area
  - DB2 for z/OS Version 9.1, Common service area storage requirements
- If you are planning to use Type 2 driver with DB2 for z/OS Version 9.1.2, ensure that DB2 for z/OS APAR PK58105 is installed.

- Review the following guidelines:
  - If the current version of WebSphere Portal and an earlier version coexist using the same DB2 for z/OS subsystem, the database user IDs for the current version of WebSphere Portal must be different than the earlier version to avoid conflicts during installation. If the two versions of WebSphere Portal connect to two different DB2 for z/OS subsystems, using the same user ID will not cause conflict.
  - Check the bufferpool allocations for your system and define the bufferpools as appropriate for your installation and define a large enough size, for example:  
    - `-db2 alter bufferpool(bp2) vpsize(15000)`  
  - Repeat for additional bufferpools as needed, for example:  
    - `bp3`
    - `bp4`
    - `bp5`
    - `bp32k1`
    - `bp32k2`
  - Update BP8K0 catalog bufferpool to 35,000 before performing database transfer. The SYSIBM.SYSDATABASE table resides in this bufferpool and is used extensively by DB2 for z/OS during database transfer.
  - Java Content Repository is a data repository that runs on DB2 to store data for applications such as Personalization and Web Content Management.
  - During database transfer from Derby to DB2 for z/OS, a supporting low order byte table space is created for the database tables storing documents. The PRIQTY and SECQTY for the table space are assigned using the default values. If you plan to store a large number of documents, you should use an automatic class selection (ACS) routine to allocate the DB2 for z/OS data sets with a primary and secondary space allocation of at least 10 cylinders, or specify a large enough value for PRIQTY and SEQTY in the DB2 DSNTIJUZ member. The table spaces can be identified by their name, having a structure like JCRDB.Sxxxxxxx; where xxxxxxx is a system-assigned combination of seven
numbers and characters.

- Also in member DSNTIJUZ, update the following parameters and then verify DSNTIJUZ runs successfully:
  
  - `edmdbdc = 204800`
  - `edmpool = 409600`
  - `edmstmtc = 204800`
  - `rruleck = yes`
  - `cachedyn = yes` (prepared, dynamic SQL statements are cached)
  - `dbacrvw = yes` (to allow database administrators to create Views)
  
- If you intend to run the LikeMinds sample, increase the `NUMLKTS` and `NUMLKUS` parameters: Ten times the default is sufficient, more depending on your usage of the sample. For example, if `NUMLKTS = 1000` and `NUMLKUS = 10000` are the installation default values, then update these values to `NUMLKTS = 10000` and `NUMLKUS = 100000`.

- Ensure that the job DSNTIJSG has been executed to create the objects required for the DB2 JDBC and ODBC metadata methods. See the DB2 Installation Guide Enabling stored procedures and tables for JDBC and ODBC support.

- Ensure that job DSNTIJMS runs successfully (re-execute binds)
- Ensure that job DSNTIJEX runs successfully

- Because large objects are stored in columns that can become very large, logging changes to these columns requires a huge amount of log space. For this reason, large object (LOB) logging is disabled by default for tablespaces that contain such data. With LOB logging disabled, you can recover full backups, but not incremental backups that can be used for point in time recovery. To recover point in time backups, you must enable LOB logging. For detailed instructions, see Technote 1306637, Managing LOB logging in DB2 for z/OS.

The database names and users on this page are suggested values and provide consistency throughout the documentation. Replace these values with values in your environment. The database name cannot exceed eight characters and can only contain letters and numbers.

If you plan to use a single DB2 for z/OS subsystem to hold data for more than one portal installation, use the same user name but a separate schema name for each database domain. For Member Manager, the user name must match the schema; the same database user cannot be used for the Member Manager databases of two distinct portal installations.

Each portal installation must be in separate and distinct WebSphere Application Server cells. If the portals are installed in the same file system, each must be installed in a separate and unique directory. If the portals are installed in different file systems, the same directory name can be used.

In a remote database environment, WebSphere Portal and DB2 Connect are installed on one machine (the local machine) and the DB2 for z/OS server is installed on a separate machine (the remote machine).

1. Review the different databases shown in the following table and replace these values with the values in your environment; schema names must be different when the database subsystem is shared. You can configure WebSphere Portal to use one database. However, using separate databases will improve scalability and performance.

<table>
<thead>
<tr>
<th>Application</th>
<th>Database name</th>
<th>Space required</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Portal Used for WebSphere Portal (at a minimum) or to hold all data. Stores information about user customizations, such as pages, and user profile and login information.</td>
<td><code>relzos commzos custzos</code></td>
<td>Depends on the number of WebSphere Portal users and portal objects, such as pages and portlets.</td>
</tr>
</tbody>
</table>
2. Review the tables and types of objects owned by each user. The architecture allows each of the following users to exist in the same database. All table spaces are approximately 2.8 GB by default. The size increases with the use of Java Content Repository.  

<table>
<thead>
<tr>
<th>Application</th>
<th>Database user placeholder</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Portal</td>
<td>releaseuscommunityusr</td>
<td>Core user who owns approximately 230 tables, used for WebSphere Portal core objects, which includes tables that store the user customizations made to pages.</td>
</tr>
<tr>
<td>Java Content Repository</td>
<td>jcr</td>
<td>Java Content Repository user who owns at least 1130 tables. The number could be higher depending on usage.</td>
</tr>
<tr>
<td>Feedback</td>
<td>feedback</td>
<td>Feedback user who owns approximately 50 tables used for logging site and personalization usage.</td>
</tr>
<tr>
<td>LikeMinds</td>
<td>likeminds</td>
<td>LikeMinds user who owns approximately 15 tables used to hold the Web site usage analysis routines and recommendation text.</td>
</tr>
</tbody>
</table>

Parent topic: Database considerations

Related Information
- Managing LOB logging in DB2 for z/OS
- WebSphere Portal detailed system requirements
- DB2 for z/OS
- DB2 Installation Guide
Planning for Oracle

When planning to transfer data to Oracle, you should consider the databases and user information, such as database names, what data is stored, and the database space needed.

Before you begin:
- Review the database considerations.
- Ensure the database that you plan to use is supported by this version of WebSphere Portal. Refer to the list of supported databases in the WebSphere Portal detailed system requirements.

The database names and users in this topic are suggested values and provide consistency throughout the documentation. Replace these values with values in your environment.

1. Review the different databases shown in the following table and replace these values with the values in your environment; schema names must be different when the database is shared. WebSphere Portal can be configured to use a single shared database if different database users are chosen for each database listed below. On Oracle, a single database configuration can significantly reduce consumption of system resources by the database management system. Using separate table spaces for all databases listed below can improve scalability and performance. Table 1. Space required for various databases

<table>
<thead>
<tr>
<th>Application</th>
<th>Database name</th>
<th>Space required</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Portal Used for the portal (at a minimum) or to hold all data. Stores information about user customization, such as pages, and user profile and login information.</td>
<td>relbcommdbcustdb</td>
<td>Depends on the number of users and portal objects, such as pages and portlets.</td>
</tr>
<tr>
<td>Personalization, Web Content Management Contains documents, personalization rules, personalization campaigns, and document library configuration information.</td>
<td>jcrdb</td>
<td>Depends on the number and size of Personalization rules and campaigns, and the number and size of items and elements created in Web Content Management.</td>
</tr>
<tr>
<td>Feedback Contains the information that is logged by your Web site for analysis of site activity and generating reports.</td>
<td>fdbkdb</td>
<td>Depends on the amount of traffic to the site. The amount of data that is logged per login-enabled page can vary.</td>
</tr>
<tr>
<td>LikeMinds Contains the recommendations to be displayed to users when their interactions with your Web site have been analyzed and predictions generated.</td>
<td>lmdb</td>
<td>Depends on the amount of traffic to the site.</td>
</tr>
</tbody>
</table>

2. Review the tables and types of objects owned by each user. The architecture allows each of the following users to exist in the same database. All table spaces are approximately 2.8 GB by default. The size increases with the use of Java Content Repository. Table 2. Tables and objects owned by database users

<table>
<thead>
<tr>
<th>Application</th>
<th>Database user placeholder</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application</td>
<td>Username</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td>releaseusrcustomizationusr</td>
<td>Core user who owns approximately 230 tables, used for WebSphere Portal core objects, which includes tables that store the user customizations made to pages.</td>
</tr>
<tr>
<td>Java Content Repository</td>
<td>jcr</td>
<td>Java Content Repository user who owns at least 1130 tables. The number could be higher depending on usage.</td>
</tr>
<tr>
<td>Feedback</td>
<td>feedback</td>
<td>Feedback user who owns approximately 50 tables used for logging site and personalization usage.</td>
</tr>
<tr>
<td>LikeMinds</td>
<td>likeminds</td>
<td>LikeMinds user who owns approximately 15 tables used to hold the Web site usage analysis routines and recommendation text.</td>
</tr>
</tbody>
</table>

**Parent topic:** Database considerations

**Related information**

- [Oracle product documentation](http://www.ibm.com/support/docview.wss?rs=688&uid=swg27007791)
Planning for Oracle RAC

When planning to transfer data to Oracle RAC, you should consider the databases and user information, such as database names, what data is stored, and the database space needed.

Before you begin:
- Review the database considerations.
- Ensure the database that you plan to use is supported by this version of WebSphere Portal. Refer to the list of supported databases in the WebSphere Portal detailed system requirements.

The database names and users in this topic are suggested values and provide consistency throughout the documentation. Replace these values with values in your environment.

The default tablespace size for Oracle RAC may need to be set to 1024MB with `autoextend` turned on for database transfer to be successful.

1. Review the different databases shown in the following table and replace these values with the values in your environment; schema names must be different when the database is shared. WebSphere Portal can be configured to use a single shared database if different database users are chosen for each database listed below. On Oracle, a single database configuration can significantly reduce consumption of system resources by the database management system. Using separate table spaces for all databases listed below can improve scalability and performance. *Table 1. Space required for various databases*

<table>
<thead>
<tr>
<th>Application</th>
<th>Database name</th>
<th>Space required</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Portal Used for the portal (at a minimum) or to hold all data. Stores information about user customization, such as pages, and user profile and login information.</td>
<td><code>relbcommdbcustdb</code></td>
<td>Depends on the number of users and portal objects, such as pages and portlets.</td>
</tr>
<tr>
<td>Personalization, Web Content Management Contains documents, personalization rules, personalization campaigns, and document library configuration information.</td>
<td><code>jcrdb</code></td>
<td>Depends on the number and size of Personalization rules and campaigns, and the number and size of items and elements created in Web Content Management.</td>
</tr>
<tr>
<td>Feedback Contains the information that is logged by your Web site for analysis of site activity and generating reports.</td>
<td><code>fdbkdb</code></td>
<td>Depends on the amount of traffic to the site. The amount of data that is logged per login-enabled page can vary.</td>
</tr>
<tr>
<td>LikeMinds Contains the recommendations to be displayed to users when their interactions with your Web site have been analyzed and predictions generated.</td>
<td><code>lmdb</code></td>
<td>Depends on the amount of traffic to the site.</td>
</tr>
</tbody>
</table>

2. Review the tables and types of objects owned by each user. The architecture allows each of the following users to exist in the same database. All table spaces are approximately 2.8 GB by default. The size increases with the use of Java Content Repository. *Table 2. Tables and objects owned by database users*
<table>
<thead>
<tr>
<th>Application</th>
<th>Database user placeholder</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Portal</td>
<td>releaseuscommunityusrcustomizationusr</td>
<td>Core user who owns approximately 230 tables, used for WebSphere Portal core objects, which includes tables that store the user customizations made to pages.</td>
</tr>
<tr>
<td>Java Content Repository</td>
<td>jcr</td>
<td>Java Content Repository user who owns at least 1130 tables. The number could be higher depending on usage.</td>
</tr>
<tr>
<td>Feedback</td>
<td>feedback</td>
<td>Feedback user who owns approximately 50 tables used for logging site and personalization usage.</td>
</tr>
<tr>
<td>LikeMinds</td>
<td>likeminds</td>
<td>LikeMinds user who owns approximately 15 tables used to hold the Web site usage analysis routines and recommendation text.</td>
</tr>
</tbody>
</table>

Parent topic: Database considerations

Related Information

- Oracle product documentation
Planning for SQL Server 2005

When planning to transfer data to SQL Server, you should consider the databases and user information, such as database names, what data is stored, and the database space needed.

Before you begin:
- Review the database considerations.
- Ensure the database that you plan to use is supported by this version of WebSphere Portal. Refer to the list of supported databases in the WebSphere Portal detailed system requirements.

The database names and users in this topic are suggested values and provide consistency throughout the documentation. Replace these values with values in your environment.

1. Review the different databases shown in the following table and replace these values with the values in your environment; schema names must be different when the database is shared. While configuring WebSphere Portal to use one database is technically possible, using separate databases can improve scalability and performance. **Table 1. Space required for various databases**

<table>
<thead>
<tr>
<th>Application</th>
<th>Database name</th>
<th>Space required</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Portal Used for the portal (at a minimum) or to hold all data. Stores information about user customization, such as pages, and user profile and login information.</td>
<td><code>reldbcommdbcustdb</code></td>
<td>Depends on the number of users and portal objects, such as pages and portlets.</td>
</tr>
<tr>
<td>Personalization, Web Content Management Contains documents, personalization rules, personalization campaigns, and document library configuration information.</td>
<td><code>jcrdb</code></td>
<td>Depends on the number and size of Personalization rules and campaigns, and the number and size of items and elements created in Web Content Management.</td>
</tr>
<tr>
<td>Feedback Contains the information that is logged by your Web site for analysis of site activity and generating reports.</td>
<td><code>fdbkdb</code></td>
<td>Depends on the amount of traffic to the site. The amount of data that is logged per login-enabled page can vary.</td>
</tr>
<tr>
<td>LikeMinds Contains the recommendations to be displayed to users when their interactions with your Web site have been analyzed and predictions generated.</td>
<td><code>lmdb</code></td>
<td>Depends on the amount of traffic to the site.</td>
</tr>
</tbody>
</table>

2. Connect at least one user to the SQL Server instance. A user can be granted permission to use several schema names, so a single user for each instance is sufficient.

3. Review the tables and types of objects owned by each schema. The WebSphere Portal architecture allows each of the following schemata to exist in the same database. All table spaces will be approximately 2.8 GB by default. The size will increase with the use of the Java Content Repository function. **Table 2. Information by application, database schema placeholder, recommended name, and function**

<table>
<thead>
<tr>
<th>Application</th>
<th>Database schema placeholder</th>
<th>Recommended name</th>
<th>Function</th>
</tr>
</thead>
</table>

123
<table>
<thead>
<tr>
<th>Application</th>
<th>Schema Name</th>
<th>Username</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Portal</td>
<td>releaseusrcommunityusrcustomizationusr</td>
<td>&lt;none&gt;</td>
<td>Core schemata. Will own approximately 130 tables for each domain. Owns WebSphere Portal core objects, which includes tables that store the user customizations made to Pages.</td>
</tr>
<tr>
<td>Java Content Repository</td>
<td>icmadmin</td>
<td>&lt;none&gt;</td>
<td>Java Content Repository schema. Will own at least 1130 tables; the number could be higher depending on usage.</td>
</tr>
<tr>
<td>Feedback</td>
<td>feedback</td>
<td>&lt;none&gt;</td>
<td>Feedback schema. Will own approximately 50 tables used for logging site and personalization usage.</td>
</tr>
<tr>
<td>Likeminds</td>
<td>LIKEMINDS</td>
<td>&lt;none&gt;</td>
<td>LikeMinds schema. Will own approximately 15 tables used to hold the web site usage analysis routines and recommendation text.</td>
</tr>
</tbody>
</table>

**Parent topic:** [Database considerations](#)

**Related information**

- [Microsoft SQL Server product documentation](http://support.microsoft.com/kb/839279)
- [Registry Entries Are Required for XA Transaction Support](http://support.microsoft.com/kb/839279)
User registry considerations

A user registry or repository authenticates a user and retrieves information about users and groups to perform security-related functions, including authentication and authorization.

User registries store user account information, such as user ID and password, that can be accessed during authentication. User repositories store user profiles and preference information. A user registry or repository is used to:

- Authenticate a user using basic authentication, identity assertion, or client certificates
- Retrieve user and group information to perform security-related administrative functions such as mapping users and groups to security roles

By default, IBM® WebSphere® Portal is installed with a federated repository with a built-in file repository. The federated repository allows you to add various user registries, realm support for Virtual Portals, and/or property extensions to create a single, working unit. The available user registries that you can add to the federated repository are LDAP user registries, database user registries, and custom user registries. **Remember:** Using the built-in file repository is not recommended in a production environment. After adding another repository and choosing the administrative users from that repository, you should remove the file repository.

Based on the federated repository, WebSphere Portal allows you to create a user base that can be federated over multiple repositories: LDAP, DB, and/or custom user registry. It also allows you to define additional attributes in a separate store if your corporate LDAP directory is read-only.

If you are using a federated repository, you must plan on where you want to store new users and groups. By default, new users and groups are stored in the default file repository. If using multiple LDAP user registries and/or database user registries, you must figure out which user registry you want to define as your default user registry where new users and groups are stored. After you add all user registries to your federated repository, you can run the `wp-set-entitytypes` task to set a specific user registry as the default location.

**Remember:** Before combining multiple user registries, review the registries for the following limitations and correct any issues:

- Distinguished names must be unique for a realm over all registries. For example, if `uid=wpsadmin,o=yourco` exists in LDAP1, it must not exist in LDAP2, LDAP3, or DB1.
- The shortname, for example `wpsadmin`, should be unique for a realm over all registries.
- The base distinguished names for all registries used within a realm must not overlap; for example, if LDAP1 is `c=us,o=yourco`, LDAP2 should not be `o=yourco`.
- Do not leave the base entry blank for any of the registries used within a realm.
- If IBM Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=groupName` and the hierarchical format is `cn=groupName,o=root`.
- The user must exist in a user registry and not within the property extension configuration; otherwise, the user cannot be a member of the realm.

If you have an application that does not support the federated repository, you can switch to a standalone LDAP user registry or a standalone custom user registry.
Overview of user registry options
IBM WebSphere Portal provides a variety of security configuration tasks. In the past, there was one task, which did not let you recover from errors or allow your user registry to meet your growing business needs. Now there are multiple tasks, which allow you to fine-tune your system to meet your business needs.

Virtual Member Manager integration
IBM WebSphere Application Server includes the Virtual Member Manager (VMM), which IBM WebSphere Portal uses to access user and group information. VMM provides an interface that enables communication between WebSphere Portal and any repository, whether federated repositories, a stand-alone repository, or your own custom user registry.

Realm support
A realm is a collection of users or groups from one or more branches of your repository tree. Those branches can be part of a single repository, for example an LDAP user registry, or it can be a combination of multiple user registries. A realm is then mapped to a Virtual Portal to allow the realm's user population to log in to the Virtual Portal. This functionality allows you to define areas within WebSphere Portal that only a limited set of users can access.

Property extension
The Property Extension, formerly known as the lookaside database, allows you to store additional user attributes into a database store without touching your backend user registry. You can use the Property Extension if your LDAP is read-only but you have a requirement that allows users to specify an additional attribute such as Timezone. You can store this additional attribute in the database store. You can also add additional attributes for an application if you cannot change your repository Schema. Property extension can be used with a federated repository, a stand-alone LDAP user registry, or a custom user registry.

Parent topic: Planning for WebSphere Portal

Related reference
Directory Search
Overview of user registry options

IBM® WebSphere® Portal provides a variety of security configuration tasks. In the past, there was one task, which did not let you recover from errors or allow your user registry to meet your growing business needs. Now there are multiple tasks, which allow you to fine-tune your system to meet your business needs.

You have the following general security options to choose from: Table 1. Security options with explanation

<table>
<thead>
<tr>
<th>Security option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP security</td>
<td>This option is a simple, single LDAP security option, which is similar to the LDAP security option provided in the past. With this option, you can create Virtual Portals with a single realm and store users and groups in a single LDAP server.</td>
</tr>
<tr>
<td>Federated security</td>
<td>This option is an evolved option to the standalone LDAP user registry with single realm support option. With this option, you can create Virtual Portals with multiple realms, you can use multiple repositories (LDAP, database, custom), and you can add Application Groups to your system. This option is good if you have to merge multiple LDAP servers into one cohesive structure. <strong>Important:</strong> You must take special care that there are no duplicate names between the various repositories. For example, if you installed the product with a Portal Administrator of &quot;wpsadmin&quot;, then the user &quot;wpsadmin&quot; should not exist in the corporate LDAP server.</td>
</tr>
<tr>
<td>Custom security</td>
<td>This option provides you with the ability to write a full controlled WebSphere Security environment by providing a Custom User Registry and a Custom Member Adapter for Virtual Member Manager (VMM). The abilities of this option will depend on your implementation.</td>
</tr>
</tbody>
</table>

Standalone LDAP security

Out-of-the-box, WebSphere Portal is configured with the default federated repository with a built-in file repository. Therefore, you must run the wp-modify-ldap-security task to switch to a standalone LDAP user registry. In order to ensure that your LDAP user registry runs properly with WebSphere Portal, you must then adapt the attribute configuration to match the configured LDAP server and your business needs. After completing the steps for these tasks, your security is ready for production.

After using your standalone LDAP user registry, you may need to manage your user registry; you can perform any of the following optional tasks to fine-tune your standalone LDAP user registry: Table 2. Optional tasks to manage the standalone user registry

<table>
<thead>
<tr>
<th>Task</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update the standalone LDAP user registry</td>
<td>You can update certain parameters such as your bind ID and password to fix issues with your LDAP user registry.</td>
</tr>
</tbody>
</table>
Federated security
Out-of-the-box, WebSphere Portal is configured with the default federated repository with a built-in file repository. The federated repository offers you the richest amount of options to meet your business needs and to allow you to easily expand your business as your needs grow. For example, if your company acquires a new business that has an existing LDAP user registry, you can just add that LDAP server to your federated repository. Choose one of the following tasks to enable a production repository: Table 3. Tasks to enable a production repository

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a federated LDAP repository to the VMM configuration</td>
<td>Select this option to add an LDAP server to the federated repository. This task does not change the current security assignment; therefore, the administrative user defined during installation is still active.</td>
</tr>
<tr>
<td>Add a federated database repository to the VMM configuration</td>
<td>Select this option to add a database to the federated repository. This task does not change the current security assignment; therefore, the administrative user defined during installation is still active.</td>
</tr>
<tr>
<td>Add a federated custom user registry</td>
<td>Select this option to add a custom user registry that your company created to the federated repository. This task does not change the current security assignment; therefore, the administrative user defined during installation is still active.</td>
</tr>
</tbody>
</table>

After you add your initial LDAP user registry, database user registry, or custom user registry, you can add additional user registries to the repository to create a multiple user registry configuration. After configuring your repository, you must remove the default file-based repository unless this is a development environment. The following tasks are required to remove the default file-based repository: Table 4. Tasks required to remove the default file-based repository

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property extension database; formerly known as the lookaside database</td>
<td>Choose this option to store additional attributes inside the VMM property extension instead of within the LDAP user registry. Some applications, such as Common Mail portlet and IBM Lotus Web Content Management use the property extension database to store additional attributes. After you enable the property extension database, you can add attributes to meet your business needs.</td>
</tr>
<tr>
<td>Create the entity type</td>
<td>Choose this option if you want to use an entity type that exists in WebSphere Portal but not within your LDAP user registry. This option creates the entity type in your user registry and adds the relative distinguished name (RDN) to map the entity type between WebSphere Portal and your user registry.</td>
</tr>
<tr>
<td>Update an existing entity type</td>
<td>Choose this option to update the default parent of an existing, single entity type; for example, if you deleted a repository and the entity type points to the deleted repository, you will need to update the information to point to a new repository.</td>
</tr>
</tbody>
</table>

After you add your initial LDAP user registry, database user registry, or custom user registry, you can add additional user registries to the repository to create a multiple user registry configuration. After configuring your repository, you must remove the default file-based repository unless this is a development environment. The following tasks are required to remove the default file-based repository: Table 4. Tasks required to remove the default file-based repository

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
</table>
After using your federated repository, you may need to manage your user registry; you can perform any of the following optional tasks to fine-tune your federated repository: *Table 5. Optional tasks to manage the federated repository*

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change the user registry where users and groups are stored</td>
<td>This task changes the default repository where new users and groups are stored.</td>
</tr>
<tr>
<td>Change WebSphere Application Server administrator</td>
<td>This task changes the WebSphere Application Server administrator user ID and password from what was defined during installation to the new user ID and password required for your clustered or standalone production environment.</td>
</tr>
<tr>
<td>Change WebSphere Portal Server administrator</td>
<td>This task changes the WebSphere Portal administrator user ID and password from what was defined during installation to the new user ID and password required for your clustered or standalone production environment.</td>
</tr>
<tr>
<td>Delete a federated repository from the VMM configuration</td>
<td>This task deleted the default file-based repository from your configuration.</td>
</tr>
<tr>
<td>Updating the federated LDAP user registry</td>
<td>Choose this option to update certain parameters such as your bind ID and password to fix issues with your LDAP user registry.</td>
</tr>
<tr>
<td>Updating the federated database user registry</td>
<td>Choose this option to update certain parameters such as the data source name, database URL, and database type to fix issues with your database user registry.</td>
</tr>
<tr>
<td>Create a new realm</td>
<td>Choose this option to create a realm, which is a group of users from one or more user registries that form a coherent group within WebSphere Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. In a federated repository, you can create multiple realms.</td>
</tr>
<tr>
<td>Property extension database; formerly known as the lookaside database</td>
<td>Choose this option to store additional attributes inside the VMM property extension instead of within the LDAP user registry. Some applications, such as Common Mail portlet and IBM Lotus Web Content Management use the property extension database to store additional attributes. After you enable the property extension database, you can add attributes to meet your business needs.</td>
</tr>
<tr>
<td>Create the entity type</td>
<td>Choose this option if you want to use an entity type that exists in WebSphere Portal but not within your LDAP user registry. This option creates the entity type in your user registry and adds the relative distinguished name (RDN) to map the entity type between WebSphere Portal and your user registry.</td>
</tr>
<tr>
<td>Update an existing entity type</td>
<td>Choose this option to update the default parent of an existing, single entity type; for example, if you deleted a repository and the entity type points to the deleted repository, you will need to update the information to point to a new repository.</td>
</tr>
</tbody>
</table>

**Parent topic:** [User registry considerations](#)
Virtual Member Manager integration
IBM® WebSphere® Application Server includes the Virtual Member Manager (VMM), which IBM WebSphere Portal uses to access user and group information. VMM provides an interface that enables communication between WebSphere Portal and any repository, whether federated repositories, a stand-alone repository, or your own custom user registry.

The Virtual Member Manager (VMM) is an abstract component within the WebSphere Application Server infrastructure. As the following diagram illustrates, WebSphere Portal uses the Portal User Management Architecture (PUMA) System Programming Interface (SPI) to retrieve and set attributes on user objects. PUMA passes these requests to VMM, which then passes the requests on to a corresponding registry adaptor that connects VMM to the repository.

The preceding diagram includes the following components:

- **Federated Repositories**
  - An out-of-box implementation of the UserRegistry interface that supports multiple repositories. To communicate with the federated repositories, both WebSphere Application Server and WebSphere Portal dispatch all operations to VMM.

- **Stand-alone LDAP**
  - An out-of-box implementation of the UserRegistry interface where a single LDAP directory is the repository. WebSphere Application Server communicates directly with the stand-alone LDAP. WebSphere Portal communicates with the stand-alone LDAP through VMM.

- **VMM SPI**
  - VMM offers a Service Provider Interface (SPI), wim.Repository, that enables communication with repositories. WebSphere Application Server uses this SPI to connect to federated repositories. WebSphere Portal uses this SPI to connect to all repositories, whether federated or stand-alone.
- **User Registry Adaptor**
  - An implementation of the VMM SPI that enables VMM to connect to a specific repository, whether an LDAP directory, database, files, or other repository. Registry adaptors enable communication between WebSphere Portal and any repository.

  **Important:** You must create a user registry adaptor if you plan to use a custom user registry or repository that WebSphere Portal does not support out-of-box. To create a user registry adaptor, implement the `wim.Repository` interface. Refer to the following topics in the WebSphere Application Server Information Center for information and instructions:
  - Repository SPI (System programming interfaces for virtual member manager adapters)
  - Sample custom adapters for federated repositories examples

**Parent topic:** User registry considerations

**Related tasks**

Setting up custom user repositories

**Related Information**

- Webcast replay of WebSphere Portal WMM to VMM comparison
- Setting up a custom user repository with Virtual Member Manager for IBM WebSphere Application Server and IBM WebSphere Portal
- IBM WebSphere Developer Technical Journal: Expand your user registry options with a federated repository in WebSphere Application Server, Using the Virtual Member Manager
- WebSphere Application Server 6.1 Information Center, VMM API
- WebSphere Application Server 6.1 Information Center, Sample custom adapters for federated repositories examples
- WebSphere Application Server 6.1 Information Center, Repository SPI (System programming interfaces for virtual member manager adapters)
Realm support

A realm is a collection of users or groups from one or more branches of your repository tree. Those branches can be part of a single repository, for example an LDAP user registry, or it can be a combination of multiple user registries. A realm is then mapped to a Virtual Portal to allow the realm's user population to log in to the Virtual Portal. This functionality allows you to define areas within WebSphere Portal that only a limited set of users can access.

For example, if you are an international company with employees in Asia, Europe, USA, and Canada, you may have an application or information that only applies to a subset of these employees. You can create a subset of employees and create a Virtual Portal that contains the application or information for that realm. Users from one realm cannot access another realm unless they are also members of that realm. For example, the wpsadmin user will not be able to log in to a Virtual Portal unless the wpsadmin user is a member of the corresponding realm.

You can create a realm that combines users from your various user registries; for example, your realm can span three LDAP user registries and a database user registry: LDAP1, LDAP2, LDAP3, and DB1. Remember: Before combining multiple user registries, review the registries for the following limitations and correct any issues:

- Distinguished names must be unique for a realm over all registries. For example, if uid=wpsadmin,o=yourco exists in LDAP1, it must not exist in LDAP2, LDAP3, or DB1.
- The shortname, for example wpsadmin, should be unique for a realm over all registries.
- The base distinguished names for all registries used within a realm must not overlap; for example, if LDAP1 is c=us,o=yourco, LDAP2 should not be o=yourco.
- Do not leave the base entry blank for any of the registries used within a realm.
- If IBM® Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is cn=groupName and the hierarchical format is cn=groupName,o=root.
- The user must exist in a user registry and not within the property extension configuration; otherwise, the user cannot be a member of the realm.

Parent topic: User registry considerations
Property extension

The Property Extension, formerly known as the lookaside database, allows you to store additional user attributes into a database store without touching your backend user registry. You can use the Property Extension if your LDAP is read-only but you have a requirement that allows users to specify an additional attribute such as Timezone. You can store this additional attribute in the database store. You can also add additional attributes for an application if you cannot change your repository Schema. Property extension can be used with a federated repository, a stand-alone LDAP user registry, or a custom user registry.

IBM® Lotus Web Content Management stores additional information for the following features:
- Web content user profiling
- Category selection trees
- Federated content elements and components

If this information cannot be stored in the main repository, for example the main repository is read-only, a property extension configuration is required.

Parent topic: User registry considerations
Security and authentication considerations

Security and authentication are key elements of a production environment. Learn about single sign-on, credential vaults and external security managers.

- **Authentication**
  Authentication requires users to identify themselves to gain access to a system or resources. The combination of a user ID and a password is the most common method of authentication. Users can identify themselves immediately upon entry to the system or the system can prompt users to identify themselves before accessing protected resources. After users successfully authenticate, the system identifies which resources-specific users have sufficient authorization to access.

- **Federal Information Processing Standards**
  Federal Information Processing Standards (FIPS) are standards and guidelines issued by the United States National Institute of Standards and Technology (NIST) for federal government computer systems. FIPS are developed when there are compelling federal government requirements for standards, such as for security and interoperability, but acceptable industry standards or solutions do not exist.

- **Planning for single sign-on**
  Single sign-on provides a secure method of authenticating a user one time within an environment and using that authentication (for the duration of the session) to access other applications, systems, and networks. In the context of IBM WebSphere Portal there are two single sign-on realms; one realm from the client to the portal and other web applications and the other realm from the portal to the backend applications.

- **Secure Socket Layer**
  Configuring IBM WebSphere Portal for SSL adds security to the client-portal exchange. It encrypts all traffic between the client browser and the server, so that no one can "eavesdrop" on the information that is exchanged over the network between the client browser and WebSphere Portal. In addition, assuming that the IBM WebSphere Application Server is also configured to accept (or even require) SSL connections, the LTPA Token and other security and session information can be completely protected against hijack and replay attacks.

- **Credential Vault**
  The Credential Vault is a service that stores credentials that allow portlets to log in to applications outside the realm on behalf of the user. It manages multiple identities for portlets and users.

- **Caching considerations**
  Information that is protected by access control and is therefore restricted to a limited set of people needs special consideration when served from an access control agnostic cache. These considerations especially apply to server side caches but you also need to consider local browser caches.

- **Planning for external security managers**
  By default, IBM WebSphere Application Server controls authentication to IBM WebSphere Portal and WebSphere Portal controls authorization (access control) to resources. You can use an external security manager such as IBM Tivoli® Access Manager or Computer Associates eTrust SiteMinder to manage authentication and authorization from a central location.

Parent topic: Planning for WebSphere Portal
Authentication

Authentication requires users to identify themselves to gain access to a system or resources. The combination of a user ID and a password is the most common method of authentication. Users can identify themselves immediately upon entry to the system or the system can prompt users to identify themselves before accessing protected resources. After users successfully authenticate, the system identifies which resources-specific users have sufficient authorization to access.

Note: You can have simultaneous, multiple log ins with the same user ID and password. However, this method can result in a non-reliable behavior depending on the client or authentication method. For this reason, IBM® WebSphere® Portal does not support simultaneous, multiple log ins.

WebSphere Portal supports the following methods for login and authentication:

- **Form-based authentication**
  - WebSphere Portal uses the IBM WebSphere Application Server Custom Form-based Authentication mechanism to prompt for identities. Users type their user ID and password in the Login portlet.

- **SSL client certificate authentication**
  - You can configure authentication with certificates that are stored in the browser or on a smart card. The certificates are stored through a Secure Sockets Layer (SSL) client certificate authentication. The authentication is done for the users when they access the protected area of the portal.

- **Third-party authentication**
  - You can also configure third-party authentication. An external security manager, such as IBM Tivoli® Access Manager, is an example. With this method the portal trusts that the authentication was done by the third-party product.

- **Automatic login with the login URL**
  - You can use the following URL which includes the user ID and password to log on: http://server:port/wps/portal/cxml/04_SD9ePMtCP1I800I_KydQvyHFUBADPmuQy?userid(userid)&password(password), where you replace the variables for server, port, userid, and password with the values set for your environment. For example, this method is suitable for automatic logon by a utility program for administrative purposes.

  **Restriction:** This URL is intended only for a single-user login. It is not intended for multiple consecutive user log ins. It is also not intended to replace the Login portlet.

**Parent topic:** Security and authentication considerations
Federal Information Processing Standards

Federal Information Processing Standards (FIPS) are standards and guidelines issued by the United States National Institute of Standards and Technology (NIST) for federal government computer systems. FIPS are developed when there are compelling federal government requirements for standards, such as for security and interoperability, but acceptable industry standards or solutions do not exist.

WebSphere Portal tolerates WebSphere Application Server's support of FIPS 140-2. WebSphere Application Server integrates cryptographic modules such as Java Secure Socket Extension (JSSE) and Java Cryptography Extension (JCE), which are FIPS 140-2 certified. Throughout the documentation and the product, the FIPS 140-2 certified IBM JSSE and JCE modules are referred to as IBMJSSEFIPS and IBMJCEFIPS, which distinguishes the FIPS-certified modules from the prior, non-certified IBM JSSE and IBM JCE modules.

The FIPS 140-2 compliant toleration means that WebSphere Portal will continue to work after WebSphere Application Server is configured to activate FIPS 140-2 compliant security modules. The WebSphere Portal product has no self-contained cryptographic support and as a result is unaware of the module differences. Functions in WebSphere Portal that use encryption include:

- Secure Sockets Layer (SSL) connections inbound from clients (but this is basically the WebSphere Application Server and HTTP Server support for SSL connections, and is transparent to WebSphere Portal)
- Internal connections between WebSphere Portal administrative functions and WebSphere Application Server administrative services (invoked, for example, when deploying a portlet which must create a Web application in WebSphere Application Server)

It is assumed, though not required, that all the connections listed above will be carried over SSL using FIPS-compliant encryption. Without FIPS 140-2 support connections may not be encrypted. And there is no requirement that every connection be SSL, even with FIPS-enabled cryptography over TLS, but again your connection may not be encrypted.

Limitations

There are some restrictions in the level of support that WebSphere Portal provides in using FIPS-certified modules:

- Lotus Sametime currently does not support FIPS 140-2.
- By default, Microsoft Internet Explorer might not have TLS enabled. To enable TLS, open the Internet Explorer browser and click Tools > Internet Options. On the Advanced tab, select the Use TLS 1.0 check box.
- The IBM Tivoli Directory Server provides the Use FIPS certified implementation option, which enables the directory server the FIPS-certified encryption algorithms uses. For information, see "Setting the level of encryption" within the IBM Tivoli Directory Server Administration Guide.
- IBMJSSEFIPS is not supported on the HP-UX platform.
- You can only use FIPS-certified JSSE providers if your servers and clients are using WebSphere Application Server.

Parent topic: Security and authentication considerations

Related tasks

Enabling FIPS
Planning for single sign-on

Single sign-on provides a secure method of authenticating a user one time within an environment and using that authentication (for the duration of the session) to access other applications, systems, and networks. In the context of IBM® WebSphere® Portal there are two single sign-on realms; one realm from the client to the portal and other web applications and the other realm from the portal to the backend applications.

Single sign-on for the client realm is established using the IBM WebSphere Application Server Lightweight Third Party Authentication (LTPA) token functionality or an Authentication Proxy. The LTPA token can also establish backend single sign-on if the backend application accepts it through the Credential Vault portlet or the Java Connector architecture.

WebSphere Portal and Java Authentication and Authorization Services

Single sign-on uses only the authentication portion of Java Authentication and Authorization Services (JAAS). WebSphere Portal builds a JAAS Subject for each logged on user. The Subject consists of Principals and Credentials. A Principal is a piece of data, such as the user ID or the distinguished name that gives the Subject's identity. A Credential is a piece of data, such as a password or a CORBA Credential that can be used to authenticate a subject. The Subject carries around the Principals and Credentials that the portlet can use directly or through the credential service.

Parent topic: Security and authentication considerations

Related information

- Lightweight Third Party Authentication (LTPA)
Secure Socket Layer

Configuring IBM® WebSphere® Portal for SSL adds security to the client-portal exchange. It encrypts all traffic between the client browser and the server, so that no one can "eavesdrop" on the information that is exchanged over the network between the client browser and WebSphere Portal. In addition, assuming that the IBM WebSphere Application Server is also configured to accept (or even require) SSL connections, the LTPA Token and other security and session information can be completely protected against hijack and replay attacks.

Configuring WebSphere Portal for SSL is a multistep process that involves configuring the following components:
- Web (HTTP) server running in front of WebSphere Application Server
- WebSphere Application Server
- WebSphere Portal

In general, the Web server must be configured to accept inbound SSL traffic. Then, the WebSphere Application Server plug-in for the Web server must be configured to forward traffic on that port to WebSphere Application Server and WebSphere Portal. This involves configuring the virtual host information. Finally, WebSphere Portal must be set up to generate self-referencing URLs using SSL as the transport. Note: This procedure might be slightly different if a front-end security proxy server such as Tivoli Access Manager WebSEAL is used. In that case, the front-end security server handles the client SSL connections. The Web server receives connections from the front-end security proxy server. Mutually authenticated SSL could be configured in the Web server and the front-end security proxy server if needed. This is highly dependent on the security requirements of each deployment.

Parent topic: Security and authentication considerations

Related Information
- *WebSphere Application Server Security Guide: Chapter 5*
Credential Vault

The Credential Vault is a service that stores credentials that allow portlets to log in to applications outside the realm on behalf of the user. It manages multiple identities for portlets and users.

Using Credential Vault, a portlet can retrieve a user's authentication identity and then pass the information to a backend application. The Credential Vault features the following two levels of sign-on:

- **Active Credentials**
  - Active Credentials allow you to establish connections through Basic Authentication, Lightweight Third Party Authentication (LTPA) token authentication or simple form-based user ID and password challenges. The Service encapsulates the single sign-on functionality for the portlet writer in an object.

- **Passive Credentials**
  - Passive Credentials retrieve stored secret data such as user ID and password or certificates. This option is more flexible but requires portlet writers to manage their own connections and authentication to backend applications with the credentials (User ID and password) they retrieved from the Credential Vault.

Credential objects can also pass IBM® Tivoli® Access Manager or Computer Associates eTrust SiteMinder single sign-on tokens to backend applications.

IBM WebSphere® Portal provides one simple database vault implementation for mappings to secrets for other enterprise applications. By default, the Credential Vault contains an administrator-managed vault segment and a user-managed vault segment. Administrator-managed vaults allow users to update mappings; however, users cannot add new applications to this vault. The user-managed vault segment allows users to add application definitions, such as a POP3 mail account, under the user vault and store a mapping there. By default, the vault uses an encryption plug-in that encodes the passwords in Base 64.

WebSphere Portal initially provides two vault adapter configurations that write to the database:
- A default vault for administrator-managed vault segments that stores credentials in the release domain: default-release
- And a default vault for user-managed vault segments that stores credentials in the customization domain: default-customization

WebSphere Portal also supports the storage and retrieval of credentials from other vault services, such as Tivoli Access Manager. WebSphere Portal ships a Credential Vault adapter for Tivoli Access Manager. This plug-in works on the following operating systems:
- AIX
- Solaris
- Windows

**Parent topic:** Security and authentication considerations
Caching considerations

Information that is protected by access control and is therefore restricted to a limited set of people needs special consideration when served from an access control agnostic cache. These considerations especially apply to server side caches but you also need to consider local browser caches.

Browser caches usually have no issues unless the computers are shared between multiple users with different levels of access. If you access WebSphere Portal from a shared computer, it is important to realize that all users who have access to the computer can access content that is cached in the local browser cache. To prevent this from happening, do not enable public or private caching of the content. Caching is disabled by default.

Depending on the type of browser you are using, you can still experience information leakage from shared computers, even if caching is completely disabled, because some browsers serve content that is accessed by clicking the browser's Back button from a separate history cache that is not affected by HTTP caching directives. As a result, if you click the Back button, you may see content generated from the previous user even if the previous user performed a logout. To prevent this from happening, the markup that is rendered on logout should explicitly clear the browser's history cache, which typically requires browser-specific script coding, or display a message to close all browser windows after logout. History cache can typically be disabled in the browser but it may be activated by default.

The WebSphere Portal and Lotus Web Content Management Performance Tuning Guide provides information about caches for WebSphere Portal and Web Content Management. After you have setup your environment, review the tuning guide to learn more about stand-alone and cluster tuning and then read about both WebSphere Portal and Web Content Management caches. WebSphere Portal Caches
Lotus Web Content Management Caches

Parent topic: Security and authentication considerations
Planning for external security managers

By default, IBM® WebSphere® Application Server controls authentication to IBM WebSphere Portal and WebSphere Portal controls authorization (access control) to resources. You can use an external security manager such as IBM Tivoli® Access Manager or Computer Associates eTrust SiteMinder to manage authentication and authorization from a central location.

**Note:** When setting up security to use an external security manager in a cluster environment and across mixed nodes, there are additional considerations. For example, you should configure your external security manager after completing all other setup tasks, including ensuring that the cluster is functional.

**External authentication**

Tivoli Access Manager and Computer Associates eTrust SiteMinder provide Trust Association Interceptors (TAI) that are used only as an authentication service. Such TAIs can be configured via the Portal configuration tasks. For example, the `enable-tam-tai` task sets up the Tivoli Access Manager TAI with the relevant properties. The Tivoli Access Manager TAI requires that the Tivoli Access Manager authorization server be available for successful single sign-on. For information about using TAI with WebSphere Application Server, refer to [http://www.ibm.com/software/webservers/appserv/was/library/](http://www.ibm.com/software/webservers/appserv/was/library/) and the WebSphere Application Server V6 Security Handbook (SG24-6316-00). For further information on the Tivoli Access Manager TAI, refer to Tivoli Access Manager Trust Association Interceptor (TAI++).

Whenever a request attempts to access a secured resource, WebSphere Application Server invokes the TAI, which validates that the request comes from a legitimate third-party authentication proxy and returns the user’s authenticated identity to WebSphere Application Server. The TAI should return either a distinguished name (DN) or a short name. WebSphere Application Server performs a registry lookup to verify the distinguished name or convert the short name to a distinguished name before searching for group memberships for that user. If the registry lookup fails, WebSphere Application Server refuses to trust the user. If the registry lookup succeeds, WebSphere Application Server generates a Lightweight Third-Party Authentication (LTPA) token for the user and stores it as a cookie for subsequent authentication during the user’s session.

A TAI is not necessary if the third-party authentication proxy provides native WebSphere Application Server identity tokens, such as a LTPA tokens. Currently, only Tivoli Access Manager WebSEAL and Tivoli Access Manager plug-in for Edge Server provide native WebSphere Application Server identity tokens. Consult the WebSEAL Administration Guide for more information about configuring Tivoli Access Manager to provide LTPA tokens. The authentication proxy determines the challenge mechanism, and WebSphere Portal relies on the authentication proxy to relay success or failure of the user identifier through the TAI or LTPA token. WebSphere Application Server sees all requests from the TAI as authenticated, but WebSphere Application Server and WebSphere Portal still perform a user and group lookup on each request. Even if the authentication proxy has successfully authenticated, WebSphere Application Server and WebSphere Portal deny access if they cannot query the user in the registry. For example, it is possible to have a user in an External Security Manager who is not accessible from WebSphere Portal because WebSphere Portal is configured to one user registry, which may not be the same registry or have the same registry configuration properties as the ESM has.

**Custom TAIs**

TAIs that allow other custom authentication services to interact with WebSphere Application Server can be written. If you use a security configuration that is different from the ones that are described in this section, you must provide and implement a TAI to communicate with the authentication proxy.
External authorization
WebSphere Portal always determines the permissions that are associated with each role, whether the role is externalized or not. Roles are always associated with a specific resource. Resources can be moved back and forth from internal to external control with the Resource Permissions portlet. Explicit role assignments are preserved when moving in both directions. However, inherited role memberships are blocked for externalized resources. When you externalize access control for a resource, the resource is administered only through the external security manager interface. After externalizing the resource, role membership must be assigned and removed using the external security manager. The Resource Permissions portlet can no longer control user access to the resource; however, the Resource Permissions portlet can move the object back to internal control.

Note the following issues:
- Private pages cannot be externalized.
- When you use the Resource Permissions portlet to externalize or internalize access control for a resource, access control for all of its public child resources moves with it. When you use the XML configuration interface (xmlaccess) to externalize or internalize access control for a resource, access control for public child resources does not change.
- After you externalize access control for a resource, you must use the external security manager to assign users to roles on the resource.
- After access control for a resource is externalized, you can use either the Resource Permissions portlet or the XML configuration interface to create additional role types on the resource. For example, suppose you create only the Administrator and Manager role types on the Market News Page. Then you externalize access control for the Market News Page. At this point, you must use the external security manager to assign users to the Administrator@Market News Page or Manager@Market News Page roles. If you decide that you want to assign users to the Editor@Market News Page role, which has not yet been externalized, follow these steps:
  1. Use the Resource Permissions portlet to create the Editor role type for the Market News Page.
  2. Use the external security manager to assign users to the Editor@Market News Page role by editing the ACL.
Remember that WebSphere Portal still determines the permissions that are associated with the externalized Editor role type.
- If a user inherits access to a resource from a parent resource, the user will lose the inherited access when the resource is externalized. If the user needs access to that resource, you must assign access through the external security manager.
- The user, who externalizes the resource, automatically receives the Administrator role on the parent resource of the externalized resource tree (if using the Resource Permissions portlet) or the resource (if using the XML configuration interface).

The decision to use an external security manager must be made with the understanding that the external security manager software’s ACL semantics override WebSphere Portal semantics. For example, if you use Tivoli Access Manager to grant anonymous membership on a role for an externally controlled portlet, you must set the ACL for that portlet to include the Tivoli Access Manager unauthenticated user group.

Note: If you use Tivoli Access Manager for authorization, you must also use it for authentication. Using Tivoli Access Manager to perform only authorization is not supported.

Planning considerations for WebSEAL junctions
A junction is an HTTP or HTTPS connection between a front-end WebSEAL server and a back-end Web application server. A junction acts as a single point of access into a Web Application Network. WebSEAL performs authentication checks on all requests for resources before passing those requests across a junction to the back-end server.

In the configuration described here, the WebSEAL component of Tivoli Access Manager handles the user authentication, and a Trust Association Interceptor (TAI) is used by WebSphere Application Server and WebSphere Portal to accept the identity of the user as asserted by WebSEAL.

To properly secure the WebSphere Application Server and WebSphere Portal system against an attack, the TAI must still authenticate the WebSEAL server, so that only requests that are legitimately presented through that WebSEAL server are
accepted. You have a choice of different ways to configure this authentication between WebSEAL and the TAI in WebSphere Application Server, depending on how much effort and performance you wish to put into securing your network. The decisions you make will determine how you set up the junctions between the WebSEAL server and WebSphere Portal. **Note:** By default, the XML configuration interface cannot access WebSphere Portal through a WebSEAL junction. To enable the XML configuration interface to access WebSphere Portal through a WebSEAL junction, use Tivoli Access Manager to define the configuration URL (/wps/config) within the junction as unprotected. Refer to the WebSEAL documentation for specific instructions about defining separate URLs within the junction and assigning separate ACLs to these URLs. After the configuration URL is defined as unprotected, only WebSphere Portal enforces access control to this URL. Other resources that are protected within the WebSEAL junction (for example, the wps/myportal URL) are still protected by WebSEAL.

### Nonencrypted junction using Basic Authentication

The identity of the user must be passed to the TAI in a header called `iv-user`, which is inserted by WebSEAL into the request that is sent from WebSEAL to the WebSphere Application Server and the WebSphere Portal servers. The junction creation option to set this up is `-c iv-user`. While WebSEAL can be configured to pass the user identity in other ways, the `iv-user` header is the only one that is supported by the TAI.

### Advanced junction configurations

For more details and options about how to configure junctions between WebSEAL with WebSphere Application Server and WebSphere Portal, including other options for specifying the WebSEAL server identity, refer to the WebSEAL Administration Guide and to the documentation for the HTTP Server that you are using with WebSphere Application Server.

The junctions between WebSEAL and WebSphere Application Server and WebSphere Portal can be configured to be encrypted or not. Encrypted junctions enhance security by making sure that no one can eavesdrop on information that is flowing between WebSEAL, WebSphere Application Server, and WebSphere Portal. However, encrypted junctions require additional administration to move the necessary signing certificates between the systems, and also have a performance cost. If you are not comfortable that your network between the firewalls is secure against unauthorized access and observation, you should use encrypted junctions between WebSEAL and WebSphere Application Server/WebSphere Portal. If you are comfortable that your network is secure against unauthorized access and observation, especially for traffic across an inner firewall, you can use unencrypted junctions between WebSEAL and WebSphere Application Server/WebSphere Portal.

#### Setting up the WebSEAL-WebSphere Application Server/WebSphere Portal Junction over SSL

requires that you configure WebSphere Application Server and the HTTP server that is used by WebSphere Application Server to accept inbound SSL traffic and route this traffic correctly to WebSphere Application Server and WebSphere Portal. This process includes importing the necessary signing certificates into at least the WebSEAL certificate keystore, and possibly also the HTTP server certificate keystore.

If you choose to use encrypted junctions between WebSEAL and WebSphere Application Server and WebSphere Portal, you can also choose to have WebSEAL identify and authenticate itself to WebSphere Application Server and the TAI using its own client-side certificate. In this case, it is possible to configure the TAI to not do any further validation of the WebSEAL server, relying on the mutual SSL connection to supply a trustable identity for the WebSEAL server.

If you choose not to use client-side certificates to identify the WebSEAL server, or if you choose not to use an SSL junction, you can identify the WebSEAL server to the TAI using a Basic Authentication (BA) header. In this case a password will be placed into the BA header, and also configured into the TAI. This represents a "shared secret" that only the TAI and the WebSEAL server know, which allows the TAI to trust that it really is the WebSEAL server that is asserting the user’s identity, and the TAI can trust it. In this case, using an SSL junction will provide additional security by protecting this BA header from observation, but the TAI will still rely on the BA header for identifying the WebSEAL server.

To set up the junction to use the Basic Authentication header to identify the WebSEAL server, use the `-b supply` option.
on the junction creation command. This will cause WebSEAL to build the BA header using the user's user ID (which is ignored by the TAI, in favor of the iv-user header) and the password that is configured into WebSEAL from the webseald-instance.conf file, on the basicauth-dummy-passwd property. The password in the webseald-instance.conf file must match the password for the ID that is specified on the com.ibm.websphere.security.webseal.loginid property of the TAI startup parameters in the WebSphere Application Server Administrative Console. For example, if you specify com.ibm.websphere.security.webseal.loginid=mistered on the TAI startup parameters, and the password for mistered is wilbur, then you must specify wilbur on the basicauth-dummy-passwd property in webseald-instance.conf on the WebSEAL server.

Parent topic: Security and authentication considerations
Cluster considerations
To increase capacity and availability, multiple portal servers can be clustered using IBM® WebSphere® Application Server Network Deployment. In a cluster the portals share a common configuration and the load is distributed evenly across all cluster instances.

IBM WebSphere Portal comes standard with WebSphere Application Server Network Deployment, a distribution of IBM WebSphere Application Server that provides a Deployment Manager server type for centrally managing and clustering a series of servers. To cluster a series of portal servers means that all portal instances share the same configuration, including database, applications, and portlets, and site design. The cluster provides a domain against which most administrative actions are performed once and synchronized with each server in the cluster. This both simplifies administration as well as ensures that all cluster members are configured and behave identically.

A server cluster also provides a shared domain in which session and cache data can be replicated and kept consistent across all members of the cluster. The cluster also provides an application synchronization mechanism that ensures consistent application management (start, stop, updates, etc.) across the cluster.

WebSphere Application Server provides an HTTP Server plug-in that can balance user traffic across all members of the cluster, and through a feature called “session affinity”, ensure that a user remains bound to a specific cluster instance for the duration of their session, to improve efficiency and performance. Additionally, in the event a cluster member is down, the workload management features of the plug-in will recognize that the instance is no longer available and will route traffic around it.

There are two types of clusters: vertical and horizontal clusters. Most large-scale deployments are mixtures of both cluster types.

It is also possible to deploy multiple portal clusters to improve availability, failover, and disaster recovery.

It is recommended that you review the cluster guidelines and limitations topics for more information about what is involved in setting up a cluster.

- Guidelines for setting up a cluster
  When planning to setup a WebSphere Portal clusters, you must take into account any cluster planning required for the WebSphere Application Server nodes. If you are not familiar with WebSphere Application Server clustering, find resources here to help you get started.

- Limitations for setting up a cluster
  The following limitations apply when setting up a WebSphere Portal cluster:

- HTTP session failover
  In a clustered environment, all requests for a particular session are directed to the same server instance in the cluster. In other words, after a user establishes a session (for example, by logging in), the user is served by the same server instance for the duration of the session. To verify which server is handling user requests for a session, you can view the global settings portlet, which displays the node name of the server handling requests. If one of the servers in the cluster fails, the request is rerouted to another server in the cluster. If distributed sessions support is enabled (either by persistent sessions or memory-to-memory session replication), the new server can access session data from the database or another server instance.

- Setting up an i5/OS database in a cluster
  To communicate with a database, servers running IBM i5/OS can use either of two JDBC drivers: the IBM Toolbox for Java JDBC driver or the IBM Developer Kit for Java JDBC driver (also referred to as the native JDBC driver). Which JDBC driver you should use depends on how you are setting up your clustered environment.
- **Security options**
  The security model in IBM WebSphere Application Server and IBM WebSphere Portal affects the planning and implementation of security in a cluster. Security is enabled by default for the WebSphere Application Server deployment manager; WebSphere Portal will not attempt to change the security settings in the deployment manager cell whenever a node is federated. This means that any existing security configuration of a stand-alone WebSphere Portal is replaced with the security settings of the deployment manager cell when it joins that cell. If you remove the node from the deployment manager cell, the original security settings are reinstated.

- **Security Scenarios**
  When setting up a cluster, there are two scenarios that must be considered. There is out-of-box security used when you first set up the cluster environment where the deployment manager has not configured the security settings. The second scenario is when an existing deployment manager has already configured the security settings prior to a node joining a cell.

- **Using external security managers in a cluster**
  If you are configuring security for IBM WebSphere Portal with an external security manager, review some additional considerations, depending on the external security manager that you are using. Perform any configuration for an external security manager after you have completed all other setup, including ensuring that the WebSphere Portal cluster is functional. In addition, review the "Systems requirement" file to ensure you are using a supported level of the external security manager software.

- **Planning for multiple clusters**
  Get an overview of the concepts associated with setting up multiple clusters. Multiple clusters are sets of servers that are managed together within a single administrative domain known as a cell, and participate in workload management.

- **WebSphere Virtual Enterprise Dynamic Clusters**
  You can create a IBM WebSphere Virtual Enterprise dynamic cluster to run IBM WebSphere Portal.

- **Cluster maintenance**
  Maintaining IBM WebSphere Portal in a cluster typically means applying corrective services (fix packs and interim fixes) or updating the software release level on each node in the cluster. Instructions for applying corrective service to a WebSphere Portal cluster are provided with the corrective service package. Before applying any maintenance, it is always important to analyze any impact to your end users and ensure that you are able to provide uninterrupted service (also referred to as 24x7 availability), even during the maintenance phase.

**Parent topic:** Planning for WebSphere Portal

**Related concepts**
Horizontal cluster topology

**Related tasks**
Setting up a cluster
Guidelines for setting up a cluster

When planning to set up a WebSphere Portal cluster, you must take into account any cluster planning required for the WebSphere Application Server nodes. If you are not familiar with WebSphere Application Server clustering, find resources here to help you get started.

Before you begin

In WebSphere Application Server, a cluster is composed of multiple identical copies of an application server. A cluster member is a single application server in the cluster. WebSphere Portal is installed as an enterprise application server within the WebSphere Application Server infrastructure. All of the clustering features available within the WebSphere Application Server infrastructure are also available and apply to WebSphere Portal. Thus, a WebSphere Portal cluster is simply a collection of multiple WebSphere Portal servers that are identically configured.

For additional planning information, refer to the appropriate IBM® WebSphere® Application Server Network Deployment version at http://www.ibm.com/software/webservers/appserv/was/library/.

Guidelines for implementing cluster environments

Cluster environments should be implemented according to the following guidelines:

- You can have a mixed 32-bit and 64-bit operating system environment; for example, you can have your deployment manager installed on a 64-bit operating system and your portal node installed on a 32-bit operating system.
- There are several approaches you can use to configure an external Web server in a clustered environment. The instructions provided here for installing a WebSphere Portal follow the approach recommended by WebSphere Application Server, which involves using the Plug-ins installation wizard to install the binary plug-in module after the cell has been set up. For a complete description of the recommended procedure for configuring an external Web server in a clustered environment, refer to the following information:
  - Windows: Selecting a Web server topology diagram and roadmap
  - UNIX: Selecting a Web server topology diagram and roadmap
  - i5/OS: Selecting a Web server topology diagram and roadmap
- The deployment manager node must be installed separately before the cells and clusters can be configured.
- WebSphere Application Server provides database session persistence and memory-to-memory replication as techniques for HTTP session failover in a clustered environment. Review the following information to determine whether you want to use one of these techniques in your cluster:
  - Windows: Task overview: Managing HTTP sessions
  - UNIX: Task overview: Managing HTTP sessions
  - i5/OS: Task overview: Managing HTTP sessions
- You can create an IBM WebSphere Virtual Enterprise dynamic cluster to run WebSphere Portal. For each node that will be part of the dynamic cluster, follow the “Setting up a cluster” instructions using a WebSphere Virtual Enterprise deployment manager. **Important for IBM i only:** Dynamic clusters are not supported because WebSphere Virtual Enterprise does not support installation on i5/OS.
  - The WasRemoteHostName and WasSoapPort properties, located in the wkplc.properties file, must be accurate at all times as many ConfigEngine scripts depend on them. If you are in a standalone environment, these parameters should point to the host name and soap port for the WebSphere_Portal application server. If you are in a clustered environment, these parameters should point to the Deployment manager host name and soap port. Only modify these properties when instructed to during the installation instructions.
- If you add a node to a cell or change a node's configuration after it has been federated to the deployment manager, synchronize the node's configuration.

- If you are planning to configure an external security manager to perform authentication or authorization for in a cluster environment, install and configure the WebSphere Portal cluster first. Verify that the cluster is working properly before proceeding with the configuration of any external security managers.

**i5/OS:** Set up a recycling procedure WebSphere Portal database with the following command; this procedure automatically removes the unused journal files:

```
CHGJRN JRN(Your DB2 DB Name/QSQJRN) DLTRCV(*YES)
```

*Your DB2 DB Name* is the value of the release.DbName property, located in the wkplc_dbdomain.properties file.

**Parent topic:** Cluster considerations
Limitations for setting up a cluster

The following limitations apply when setting up a WebSphere Portal cluster:

- You must install WebSphere Portal as a standalone node before creating a cluster; you can no longer install WebSphere Portal into a managed node.
- Except for the temporary state during the initial setup of the cluster, WebSphere Portal is not supported when running on a managed node that is not part of a clustered environment. **Note:** A cluster can be created which contains only one WebSphere Portal server, enabling a single WebSphere Portal server to be operational in a managed cell.
- In a clustered environment, it is not possible to change settings through the Global Settings portlet or the XML configuration interface. These changes must be made by modifying the respective properties in the WebSphere Application Server administrative console.
- To support search in a clustered environment, you must install and configure search for remote search service on an application server node that is not part of the cluster.
- Administrative actions for WebSphere Portal are immediately visible for the user who performs them. However, another user can be assured of seeing the changes only if the user logs out of WebSphere Portal and then logs back in. This limitation applies to both cluster and non-cluster environments.
- When creating a cluster or a cluster member, do not use spaces in the cluster name or the cluster member name.
- **IBM® i5/OS limitations:** Installation in a mixed node environment is not supported in an i5/OS environment.
- Dynamic clusters using IBM® WebSphere® Virtual Enterprise are not supported.
- For the deployment manager and each WebSphere Portal node to be in the cluster, verify that each system clock is set to within 5 minutes of the others or the `addNode` command will fail.

**Parent topic:** Cluster considerations
HTTP session failover

In a clustered environment, all requests for a particular session are directed to the same server instance in the cluster. In other words, after a user establishes a session (for example, by logging in), the user is served by the same server instance for the duration of the session. To verify which server is handling user requests for a session, you can view the global settings portlet, which displays the node name of the server handling requests. If one of the servers in the cluster fails, the request is rerouted to another server in the cluster. If distributed sessions support is enabled (either by persistent sessions or memory-to-memory session replication), the new server can access session data from the database or another server instance.

Distributed session support must be configured separately in WebSphere Application Server. Refer to the WebSphere Application Server documentation for information:

- Windows and UNIX: Distributed sessions
- i5/OS: Distributed session support

By default failover support is available for WebSphere Portal and any portlets that are installed with the product. To take advantage of failover support with your own developed portlets, you must ensure that your portlets are implemented according to best practices.

Failover and lost data

Data that is stored within the JVM memory and not managed by the application server or the WebSphere Portal server for replication might be lost in the case of failover. Even with the distributed session support, during a failure, users will not be able to recover any uncommitted information that is not stored in sessions or other replicated data areas (such as a distributed Map or render parameters). In such cases, users might have to restart a transaction after a failover occurs. For example, if you are in the process of working with a portlet and have navigated between several different screens when a failover occurs, you might be placed back at the initial screen, where you would need to repeat your previous steps. Similarly, if you are attempting to deploy a portlet when a failover occurs, the deployment might not be successful, in which case you must redeploy the portlet. The validity of user login sessions is maintained despite node failures with distributed session support enabled.

In cases where a portlet does not support failover, a “Portlet Unavailable” message is displayed for the portlet after a failover occurs. If a portlet supports partial or incomplete failover, some data displayed by the portlet before the failover could disappear after the failover occurs, or the portlet might not work as expected. In such extreme cases, the user should log out and log back in to resume normal operation.

After a failover occurs, the request is redirected to another cluster member by the Web server plug-in. Most browsers will issue a GET request as a response to a redirect after submitting a POST request. This ensures that the browser does not send the same data multiple times without the user's knowledge. However, this means that after failover, users might have to refresh the page in the browser or go back and resubmit the form to recover the POST data.

Note: Any portlets or applications that use POST data are affected by this behavior.

Failover considerations for the Web Content Authoring Portlet

When configuring distributed session support in WebSphere Application Server, either for persistent sessions or memory-to-memory session replication, you can configure the Custom tuning parameters setting to determine which session attributes are replicated and how often the replication takes place. You can select a tuning level from “Very high” to optimize for performance to “Low” to optimize for failover.
In order for session information to be preserved after failover when using the Web Content Authoring Portlet, the custom tuning level should be set so that all session attributes are written.

For example, if the write frequency is set as "Time-based" with a frequency of 10 seconds, any changes to the Web Content Authoring Portlet session within 10 seconds of the failover will be lost. If the write frequency is set as "End of the servlet service", the Web Content Authoring Portlet session will remain intact after failover.

During a failover condition, a Web Content Management user might be placed back at the initial navigation screen, might have to refresh the browser window. Uncommitted data that is not stored in sessions or other replicated data areas may be lost. However, apart from these issues, there should be no loss of service and the user should be able to continue working.

Parent topic: Cluster considerations
Setting up an i5/OS database in a cluster

To communicate with a database, servers running IBM® i5/OS can use either of two JDBC drivers: the IBM Toolbox for Java JDBC driver or the IBM Developer Kit for Java JDBC driver (also referred to as the native JDBC driver). Which JDBC driver you should use depends on how you are setting up your clustered environment.

The JDBC driver is specified by the `db2_iseries.DbDriver` property in the `wkplc_dbtype.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory. You can specify the value by editing the file manually or by selecting the appropriate value using the configuration wizard.

- **Native JDBC driver**: `com.ibm.db2.jdbc.app.DB2Driver`
- **IBM Toolbox for Java JDBC driver**: `com.ibm.as400.access.AS400JDBCDriver`

### Scaling topology considerations

Vertical and horizontal scaling topologies in an i5/OS environment require different JDBC driver configurations, according to how you deploy your database.

<table>
<thead>
<tr>
<th>Scaling topology</th>
<th>JDBC driver considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertical scaling</td>
<td>When setting up a vertical cluster, you can install the database locally on the same machine as your portal or remotely on a separate machine. Use the appropriate JDBC driver, depending on where the database is installed. <strong>Local database:</strong> Either the native JDBC driver or the IBM Toolbox for Java JDBC driver can be used. <strong>Remote database:</strong> The IBM Toolbox for Java JDBC driver must be used for connections to a remote database.</td>
</tr>
<tr>
<td>Horizontal scaling</td>
<td>When setting up a horizontal cluster, you must use the IBM Toolbox for Java JDBC driver. The typical configuration is to use a remote database for primary and secondary nodes in the cluster. If you choose, you can use a local database for the primary node and configure the secondary nodes to use that database, just as you would any other remote database. However, regardless of whether you choose to include a local database in your environment, you must use the IBM Toolbox for Java JDBC driver with your horizontal cluster.</td>
</tr>
</tbody>
</table>

### Using a local database in an i5/OS horizontal cluster

Although the instructions for setting up a horizontal cluster describe how to use a remote database for both primary and secondary nodes, you can choose to configure your i5/OS horizontal cluster to use a local database for the primary node instead.
Note: Although it is possible to use a local database on a secondary node instead of the primary node, this configuration has not been tested and is not documented here.

Important: Even though you are using a local database for the primary node in this scenario, all database connections are configured as if the database were remote. Specifically, you must use the IBM Toolbox for Java JDBC driver (com.ibm.as400.access.AS400JDBCDriver) when configuring the database for both primary and secondary nodes.

To use a local database with your primary node, perform the database configuration, with the following variations when updating properties files in the <code>wp_profile_root/ConfigEngine</code> directory.

- **wkpic_dbtype.properties**
  - Specify the JDBC driver in the <code>db2_iseries(DbDriver</code> property. For example:
    
    ```
    db2_iseries(DbDriver=com.ibm.as400.access.AS400JDBCDriver
    ```
  - Specify the database location as remote in the <code>db2_iseries(DbDriverType</code> property. For example:
    
    ```
    db2_iseries(DbDriverType=4
    ```

- **wkpic_comp.properties**
  - Specify the primary node's host name for the <code>domain(DbName</code> properties. For example: <code>release(DbName=primary_host_name/wpsdb</code>
  - Specify the primary node's host name in the <code>domain(DbUrl</code> properties. For example: <code>release(DbUrl=jdbc:as400:primary_host_name/wpsdb</code>

Note: If using the configuration wizard for database transfer, update the values in the wizard panels rather than in the properties files.

Complete all other configuration as described. When configuring secondary nodes in this scenario, perform your database configuration as you would for any remote database, using the primary node's host name for the database transfer.

Parent topic: Cluster considerations
Security options
The security model in IBM® WebSphere® Application Server and IBM WebSphere Portal affects the planning and implementation of security in a cluster. Security is enabled by default for the WebSphere Application Server deployment manager; WebSphere Portal will not attempt to change the security settings in the deployment manager cell whenever a node is federated. This means that any existing security configuration of a stand-alone WebSphere Portal is replaced with the security settings of the deployment manager cell when it joins that cell. If you remove the node from the deployment manager cell, the original security settings are reinstated.

Default security settings
The default security that is enabled on the deployment manager profiles and WebSphere Portal profiles installation is the Virtual Member Manager (VMM) federated security with a single file-based repository configured. If you plan to add the standalone node into a deployment manager cell, there is no need to modify this default security setting on a WebSphere Portal node when the purpose of that node is to join a deployment manager cell and run as part of a cluster. During federation, the standalone environment security settings are replaced with the deployment manager security settings. The original standalone environment security settings are preserved and will revert back to the original settings if you remove the node from the cluster. **Note:** If administrative security is deselected during installation of the deployment manager or is disabled after the deployment manager is installed, it must be enabled prior to executing the security configuration tasks on the WebSphere Portal cluster members.

Security options for a cluster
There are many security options that can be used in a cluster. All of the VMM federated security options, including multiple LDAP repositories, database repositories, and the default file-based repository can be used. Additionally there is an option to use standalone LDAP security instead of the VMM federated security approach. WebSphere Portal provides a number of security tasks, which can be used to modify the WebSphere Application Server security settings and make the required updates to the WebSphere Portal configuration in a single step. As soon as a WebSphere Portal node is federated into a deployment manager cell, all executed WebSphere Portal security tasks will update the security configuration on the deployment manager cell. Run security tasks after federating the WebSphere Portal node because the Deployment Manager cell does not contain the configuration resources required to run the security tasks.

The tasks under “Setting up a clustered production environment” recommend configuring security before configuring your additional nodes. If you configure your security after configuring your additional nodes or if you need to update your security configuration after you created your clustered environment, you will need to run an additional task to update the security settings on the secondary nodes; see “Configuring security after cluster creation” for information. **Note:** It is not recommended to use the file-based repository in a production environment. The reason is that updates are only possible through the WebSphere Application Server administrative console, not through portal user management. These updates are sent to each node in the cell using deployment manager file synchronization. This can be time consuming for large volumes of users and groups. Also, synchronization does not occur at the same time for all nodes in a cell, so there will be time windows when the nodes in the cell have differing security definitions.

Parent topic: Cluster considerations
Related concepts
Security options

Related tasks
Enabling LDAP security after cluster creation
Security Scenarios

When setting up a cluster, there are two scenarios that must be considered. There is out-of-box security used when you first set up the cluster environment where the deployment manager has not configured the security settings. The second scenario is when an existing deployment manager has already configured the security settings prior to a node joining a cell.

**Out-of-box security**

The first scenario is when the default Virtual Member Manager (VMM) file-based repository security is used on both the WebSphere Portal nodes and the deployment manager. When the WebSphere Portal node is federated into the deployment manager cell, the node's security settings are replaced with the deployment manager's security settings. Thus, prior to federating the first WebSphere Portal node into the cell, the required group for WebSphere Portal administrators and administrative user; for example, wpsadmins and wpsadmin; must be defined in the deployment manager's security repository. Otherwise, the WebSphere Portal administrators group and administrative user will be lost when federating the node into the deployment manager.

Once the cluster has been set up, you can modify the security settings of the cell. Although it is possible to modify security in the cell using the IBM® WebSphere® Application Server Administrative Console, you should use the WebSphere Portal security tasks to change cell security in order to ensure that the security configuration settings for WebSphere Application Server and WebSphere Portal are identical.

**Note:** Using the WebSphere Application Server Administrative Console to configure or update the out-of-box security is NOT supported in a stand-alone environment. This is only supported in a clustered environment that uses the Deployment Manager Administrative Console.

**Modified security with Virtual Member Manager (VMM) federated**

The second scenario is when the existing deployment manager cell has already modified its default security setting prior to the first WebSphere Portal node joining the cell. WebSphere Portal supports the capability of using two different sets of administrative user ID and password credentials when federating a WebSphere Portal node into a cell – one set for the WebSphere Portal node authentication and one set for deployment manager authentication. This means that it is not necessary to define a common administrative user ID before WebSphere Portal joins the cell. If the deployment manager cell is using federated VMM with additional repositories, the security settings on the Portal node are replaced with the modified deployment manager VMM federated security settings. The original stand-alone environment security settings are preserved and will revert back to the original settings if you remove the node from the cluster.

**Modified security with standalone Lightweight Directory Access Protocol (LDAP) server**

If the deployment manager cell is using standalone LDAP security, it is necessary to configure the LDAP values into the WebSphere Portal property files before federation. This enables WebSphere Portal to dynamically adapt to the existing standalone LDAP security settings of the cell. As with the first scenario, once the cluster has been set up then security changes to the deployment manager cell security settings can be made using the WebSphere Portal security tasks, and additional WebSphere Portal nodes may be added to the cell following the same procedures.

Parent topic: [Cluster considerations](#)
Using external security managers in a cluster

If you are configuring security for IBM® WebSphere® Portal with an external security manager, review some additional considerations, depending on the external security manager that you are using. Perform any configuration for an external security manager after you have completed all other setup, including ensuring that the WebSphere Portal cluster is functional. In addition, review the "Systems requirement" file to ensure you are using a supported level of the external security manager software.

- Prerequisites
  - System requirements

General considerations

The following considerations apply to all external security managers:

- When setting up security in a cluster to use an external security manager, ensure that you perform the security configuration on each node in the cluster, as described in the following topics:
  - IBM Tivoli® Access Manager: Configuring Tivoli Access Manager
  - Computer Associates eTrust SiteMinder: Configuring eTrust SiteMinder for authentication and authorization
  - If you make any changes to the external security manager configuration after initially setting it up, first make the changes in the wkplc_comp.properties on the primary node of the cluster. If additional nodes exist in the cluster, ensure that any changes you make to the wkplc_comp.properties file on the primary node are propagated to the wkplc_comp.properties file on other nodes in the cluster.

Tivoli Access Manager considerations

- Ensure that you run the validate-pdadmin-connection task on each node in the cluster.

- If the validate-pdadmin-connection task fails, run the run-svrsssl-config task before attempting to run validate-pdadmin-connection again. Note that the wp.acc.impl.PDServerName parameter in the wkplc_comp.properties file represents an individual configured AMJRTE connection to Tivoli Access Manager, and each node in the cluster must have a unique value for wp.acc.impl.PDServerName before running the run-svrsssl-config task.

- If you are using an external Web server, additional configuration is required before running any task to configure an external security manager with a WebSphere Portal cluster. Edit the wkplc_comp.properties file on each node, and ensure that the values for the wp.ac.impl.JunctionHost and wp.ac.impl.JunctionPort properties are set to the backend server host name and port number you are using for your Web server.

- Ensure that the WebSEAL Trust Association Interceptor (TAI) parameters are the same on each node in the cluster. If you run a configuration task at a later time that overwrites the WebSEAL junction, the WebSphere Application Server TAI properties are not automatically updated, so you must manually ensure that all nodes are using the same parameters.

- Note the file location specified by the wp.ac.impl.PDPermPath parameter in the wkplc_comp.properties file. This property indicates the location of the Tivoli Access Manager AMJRTE properties file (PdPerm.properties). In a cluster composed of nodes with different operating systems, the location of the PdPerm.properties file might differ, depending on the node. You must ensure that the value of the wp.ac.impl.PDPermPath property on each node corresponds to the location of the PdPerm.properties file. This value can be set globally for all cluster members by using the configURL property, accessed in the deployment manager administrative console. Depending on your version of WebSphere Application Server, make the appropriate selection from the navigation:
- For WebSphere Application Server Version 7.0: Click Security > Global security > Java Authentication and Authorization Service > Application logins > Portal_Login > com.tivoli.mts.PDLoginModule > Custom properties.

To ensure that the location of the PdPerm.properties file is properly specified, use one of the following approaches:
- If your nodes are all on UNIX platforms, use the UNIX link command (ln) to ensure the value for configUrl resolves on each node.
- If the configURL parameter is not set in the administrative console, the default location is relative to the JAVA_HOME system property under the following path: java_home/jre/PdPerm.properties. Make sure the wp.ac.impl.PDPermPath parameter for each node is set to this relative location before running the run-svrssl-config task, and completely remove the configURL property from the PDLoginModule custom properties.

Parent topic: Cluster considerations
Planning for multiple clusters

Get an overview of the concepts associated with setting up multiple clusters. Multiple clusters are sets of servers that are managed together within a single administrative domain known as a cell, and participate in workload management.

IBM® WebSphere® Application Server Network Deployment provides the ability to manage many application servers and application server clusters within a single administrative domain, or cell. The single cell has the following advantages:

- A single administration User Interface (Administrative Console)
- A single administrative scripting client (wsadmin)
- Shared resources at the cell, node, or server scope
- Replication domains for sharing application data, state information, and caches
- Work load management at the Web server level providing a single server identity for all applications hosted across the cell, enabling ease of collaboration between applications while building a rich end-user application experience

An administrator’s goal is to manage as many WebSphere Portal and portal-based products within the same managed cell as possible, to take advantage of these administrative and runtime features.

WebSphere Portal provides the ability to federate multiple, independently configured portals into the same cell. While there are limitations to this support, this allows multiple clusters to be managed together, where one portal may be providing different applications or services than another portal. With a common server identity through the Web server, these services and applications can integrate seamlessly at the browser through the latest in Web 2.0 technology (for example through the use of Ajax and REST services).

How multiple clusters work in a single cell

It is important to first understand that a cell’s configuration has the notion of scope, which controls the visibility of that resource to other resources and application server instances. An example of a resource might be a data source definition, or a WebSphere variable definition. Scopes are typically defined as being one of the following:

- **Cell**
  - All resources defined at this scope are visible to all other resources defined in the cell, and are thus configured globally available

- **Node**
  - A cell has one or more nodes, and each node is named and matches with some WebSphere Application Server profile on some physical server. All resources defined at this scope are visible only to other resources defined in this same node, including any server definitions

- **Server**
  - A node has one or more server definitions. All resources defined at this scope are visible only to that server. No other server or node can make use of these resources

- **Cluster**
  - A resource defined at a cluster scope is visible to all cluster members, or server instances, in this cluster, but is not visible to any other servers in the same nodes
Note: Resources defined within a circle, in the above diagram, can be seen by all other resources defined within that circle and any other scopes defined within that circle.

Within this concept of scope, an important point is that all enterprise applications are cell-scoped. In other words, there can only be one enterprise application with a given name in the cell. If multiple servers and clusters, or multiple clusters require the use of that enterprise application, they must share it. Note: IBM WebSphere Virtual Enterprise offers the ability manage multiple editions of the same enterprise application, including the mapping of these editions to different servers and clusters, or to different clusters. WebSphere Portal, however, does not currently exploit this feature.

Typically, when installing an enterprise application that will be shared across multiple clusters, the administrator simply installs the enterprise application archive (EAR) into the cell’s management server, Deployment Manager, and then maps the application to the target clusters where it will run. Since WebSphere Portal installs several enterprise applications as part of its basic configuration and typically before any cluster is defined, special steps have to be followed to ensure that these infrastructure applications are appropriately shared when multiple WebSphere Portal clusters are defined within the same cell. And by extension, since these are infrastructure applications, all WebSphere Portal-based clusters must be at the same version.

Since portlets are enterprise applications of a special type, it is possible, but not always appropriate, to share portlets across multiple clusters. Many portlets (for example WebSphere Portal administration) are considered part of the infrastructure, and as a result can be shared across multiple clusters. Most end-user application portlets will be specific to certain clusters and will be installed as such. See the Portlet deployment best practices section for more details.

Also, the J2EE security configuration for the cell is shared by all servers and clusters managed in the cell. Therefore, each server and cluster must share the same underlying user repository against which users are authenticated when using any application hosted by any server and/or cluster in that same cell.

To summarize at a high level, supporting multiple clusters in the same cell involves:

- A common security model, including user repositories, for every cluster
- Some number of common enterprise applications and portlets, that must be made common as part of the federation and clustering process
- Installing portlets into certain clusters, or across clusters, as appropriate
- Understanding how to tell if enterprise applications are shared between clusters
- Defining other resources at the appropriate scope, depending on the usage goals
Limitations

- All portal clusters must be at the same maintenance levels
  - Because WebSphere Portal is made up of several enterprise applications, and because these applications are tightly coupled to the underlying services and infrastructure, all portal-based clusters in the same cell must be at the same service level.

- Process Server considerations
  - In the case where multiple clusters need access to a common WebSphere Process Server, the server should be centralized within its own cluster, and the client install option of WebSphere Process Server should be used in conjunction with WebSphere Portal to allow remote access to the central process server cluster.

- WebSphere Virtual Enterprise Static clusters
  - You cannot install WebSphere Virtual Enterprise on i5/OS; therefore, you must install WebSphere Virtual Enterprise on a remote non-i5/OS machine and use the WebSphere Virtual Enterprise On Demand Router (ODR) to balance workload across multiple clusters. See Routing requests across clusters for information about ODR.

- Database sharing between multiple clusters
  There are multiple database domains that are used in a typical IBM WebSphere Portal environment, such as the release domain, customization domain, and community domain. In most cases, the relevance of data within these domains is WebSphere Portal specific. In other words, different database domains are used for different configurations, because the application mix and user community is likely different. In many cases, you may want to support multiple identically configured WebSphere Portal installations in the same cell, where many of the database domains might be shared, for ease of maintenance or failover.

Parent topic: Cluster considerations

Related concepts

Understanding configuration options for process integration
Database sharing between multiple clusters

There are multiple database domains that are used in a typical IBM® WebSphere® Portal environment, such as the release domain, customization domain, and community domain. In most cases, the relevance of data within these domains is WebSphere Portal specific. In other words, different database domains are used for different configurations, because the application mix and user community is likely different. In many cases, you may want to support multiple identically configured WebSphere Portal installations in the same cell, where many of the database domains might be shared, for ease of maintenance or failover.

In the case that two independently configured clusters exist within the same cell, each cluster should have its own set of database domains. In the case that two clusters configured identically exist within the same cell, all database instances should be shared except the release and JCR database domain. This ensures that all user-specific and community data is shared between clusters, while each cluster’s static configuration can be independently updated.

Special Database Considerations when Sharing Domains

When configuring multiple clusters which will reside in the same IBM WebSphere Application Server cell, special attention must be given to the database settings.

- Based on your configuration case, determine which database domains you want to share with other clusters which reside in the same cell (multiple cluster environment). **Important:** JCR domains and release domains cannot be shared between different Web Content Management clusters or servers. Each distinct cluster or server in your Web Content Management system must use a separate JCR or release domain. For example:

<table>
<thead>
<tr>
<th>Development Server</th>
<th>Authoring Cluster</th>
<th>Staging Server</th>
<th>Delivery Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCR Domain 1</td>
<td>JCR Domain 2</td>
<td>JCR Domain 3</td>
<td>JCR Domain 4</td>
</tr>
<tr>
<td>Release Domain 1</td>
<td>Release Domain 2</td>
<td>Release Domain 3</td>
<td>Release Domain 4</td>
</tr>
</tbody>
</table>

- Assign the data source names for the domains based on which databases should be shared between the clusters and which should be unique per cluster. A single data source cannot be used for multiple domains if the domains are a mixture of shared and non-shared.
- Maintain the same number of data sources with identical names in the case where enterprise applications are to be shared across all clusters in the same cell, so that data source bindings in the applications can be resolved on every cluster in which they run.
- When installing the primary node of the next cluster (Cluster B), the node can be configured to use the shared database domains by setting the appropriate property values in the `wkplc_comp.properties` file.

**Parent topic:** Planning for multiple clusters
WebSphere Virtual Enterprise Dynamic Clusters

You can create a IBM® WebSphere® Virtual Enterprise dynamic cluster to run IBM WebSphere Portal.

**Important for i5/OS only:** Dynamic clusters are not supported because WebSphere Virtual Enterprise does not support installation on IBM i5/OS.

For each node that will be part of the dynamic cluster, follow the instructions to install and configure WebSphere Portal in a production environment using WebSphere Virtual Enterprise as the deployment manager. However, do not run the task to set up a static cluster (cluster-node-config-cluster-setup). After installing and preparing all nodes, follow the instructions provided to set up a WebSphere Virtual Enterprise node group and dynamic cluster for WebSphere Portal.

The WebSphere Virtual Enterprise On Demand Router (ODR) component provides capabilities such as workload balancing, prioritization, health monitoring, and dynamic operations for dynamic clusters. An ODR can be configured to provide multi-cluster routing, including dynamic clusters located in remote cells, and routing to other servers that are not running WebSphere Virtual Enterprise. The ODR can serve as a replacement for the HTTP server plug-in, but in many configurations both components are used. The HTTP server could be located in the demilitarized zone to serve static content and to provide an entry point to the private network where the ODR resides.

Review the following considerations before configuring the On Demand Router (ODR) to route traffic to WebSphere Portal clusters:

- Internal users can send requests directly to the ODR instead of through a front-end web server. When sending direct requests, you must configure the ODR to append a via header to the HTTP requests. Set the value of the ODR custom property `http.compliance.via` to `true`; see the "On demand router settings" link below for information. **Note:** This step is not required when sending user traffic through the web server to the ODR because the web server appends the via header to the HTTP request.

- The ODR can selectively route traffic to clusters based on the incoming URL. You can configure IP alias values for the ODR and then define routing rules to associate user traffic for each IP alias to the appropriate WebSphere Portal cluster.

- You can use the ODR to load balance traffic among identical portal clusters. You can configure a Multicluster Routing Policy (MCRP) for the ODR to identify the destination clusters and the type of load balancing; see the "Configuring the on demand router for multi-cluster failover and load balancing routing" link below for information. **Note:** If you are configuring the ODR to route traffic to remote portal static clusters using Generic Server Cluster definitions, the `cell_name` value used by the MCRP policy needs to be the local cellname where the ODR resides and not the remote cell where the portal cluster resides.

- You can also use the ODR to route traffic to remote portal clusters, both static and dynamic, by defining a generic server cluster for each target portal cluster; see the "Defining generic server clusters for remote ODR cells" link below for information. **Note:** If you are routing to remote static clusters that use vertical cluster members, you must perform the optional step at the end to define a server custom property for each port in the generic server cluster.

**Important:** When applying maintenance to upgrade the WebSphere Virtual Enterprise and WebSphere Application Server Network Deployment version level, it is important that the deployment manager remain inactive until both upgrades are complete because if the deployment manager is active before both upgrades are complete it may detect an incompatible version of WebSphere Virtual Enterprise and remove some required resources from the dynamic cluster. Keeping the deployment manager inactive until both Network Deployment and WebSphere Virtual Enterprise updates have been completed will insure that this potential problem does not occur.

See Preparing WebSphere Application Server Network Deployment to install the product for information on the order of product installation to prepare for WebSphere Virtual Enterprise and ODR.
See Creating and configuring ODRs for information on creating an ODR.

**Parent topic:** Cluster considerations

**Related information**
- Defining generic server clusters for remote ODR cells
- Configuring the on demand router for multi-cluster failover and load balancing routing
- On demand router settings
Cluster maintenance

Maintaining IBM® WebSphere® Portal in a cluster typically means applying corrective services (fix packs and interim fixes) or updating the software release level on each node in the cluster. Instructions for applying corrective service to a WebSphere Portal cluster are provided with the corrective service package. Before applying any maintenance, it is always important to analyze any impact to your end users and ensure that you are able to provide uninterrupted service (also referred to as 24x7 availability), even during the maintenance phase.

For the discussion in this section, we classify fixes as “minor” if they do not update the underlying WebSphere Portal databases or require version upgrades to other supporting software such as databases servers or IBM WebSphere Application Server. Most of the WebSphere Portal service packs are not considered minor and may require the use of a separate installation procedure to ensure 24x7 availability.

**Note:** If you have not implemented horizontal scaling in your environment; for example, you have only vertical servers in your cluster, any fix that requires a restart will result in temporary outage for your end users. Existing 24x7 install procedures do not apply to these environments.

### Minor fixes

All minor fixes to WebSphere Portal in a clustered environment can be deployed by simply applying the fix on each cluster node using the install instructions supplied with the fix. You do not need to remove the node from cluster to apply minor fixes; doing so may result in the inability to add the node back to the cluster. When applying minor fixes that might update previously deployed enterprise applications, be sure to turn off the auto-synchronization feature of the deployment manager before applying the fix. After the fix is deployed on all cluster nodes, you can force a manual synchronization using the deployment manager to ensure that all updates are synchronized on the nodes. You can then enable the auto-synchronization feature again.

If the documentation associated with the minor fix requires that WebSphere Portal or WebSphere Application Server be restarted, be sure to apply the minor fix one node at a time. This will enable other nodes to continue to provide service to your end users. However, if the fix requires an update to the WebSphere Portal databases, you might be required to stop the cluster before applying the fix. If this is the case, use a procedure that ensures 24x7 availability.

### Service packs

There are multiple approaches to installing service packs into a WebSphere Portal clustered environment. The recommended approach uses multiple production clusters. Installing a service pack involves modifying the IP sprayer to remove a cluster from receiving user requests, which allows time to finish handling existing user sessions, and upgrading that cluster to install the service pack on all nodes. After verification testing assures that the upgraded cluster is operational, it can start receiving production traffic while the next cluster is taken offline and goes through the upgrade process. This approach preserves complete 24x7 availability during the upgrade process.

A separate document is available which describes the process of installing WebSphere Portal service packs (fix packs) into an existing cluster while maintaining 24x7 availability. To briefly summarize this procedure, you remove a node or set of nodes from the flow of user traffic by configuring the IP sprayer and Web server. You then upgrade the node with the service packs. After upgrading is complete, you return the node or set of nodes to the flow of user traffic, while repeating the procedure with the next node or set of nodes. This process continues until you have upgraded all nodes in the cluster.

**Important:** While the upgrade process is taking place, some portlets may become temporarily unavailable because of updates to the shared database, which are incompatible with the previous version of the portlet. This can introduce functional limitations to the 24x7 availability when using this process.
Parent topic: Cluster considerations
WebSphere Process Server integration

IBM® WebSphere® Portal integrates business processes and related tasks that are managed by IBM WebSphere Process Server. Using process integration, you can manage human tasks for business applications within the portal environment to support service-oriented architecture (SOA). Portals that provide the interface for human tasks in business processes that are managed by a process server are referred to as process portals. Successful deployment of process portals requires planning for installation, configuration, and use within your enterprise.

1. Prerequisites for setting up process portals
   IBM WebSphere Portal does not include the Business Process Choreographer component of IBM WebSphere Process Server. To support business processes in your portal environment, install the WebSphere Process Server Client on the WebSphere Portal server and connect to a remote WebSphere Process Server.

2. Understanding configuration options for process integration
   Review the configuration scenarios to determine which one meets the needs of your enterprise for SOA process portal support with full-scale business process integration. You can configure the IBM WebSphere Portal server, either a stand-alone server or a cluster, and the IBM WebSphere Process Server server, either a stand-alone server or a cluster, in different cells of IBM WebSphere Application Server. Alternatively, you can configure stand-alone or clustered portal servers and process servers in the same cell of IBM WebSphere Application Server.

3. Planning to use process portals
   Consider how you will use business process integration in your portal environment.

Parent topic: Planning for WebSphere Portal

Related tasks
Integrating with WebSphere Process Server

Related information
- IBM WebSphere Process Server library
- IBM WebSphere Process Server, Version 6.1.0, documentation
- IBM Business Process Management
Prerequisites for setting up process portals

IBM® WebSphere® Portal does not include the Business Process Choreographer component of IBM WebSphere Process Server. To support business processes in your portal environment, install the WebSphere Process Server Client on the WebSphere Portal server and connect to a remote WebSphere Process Server.

The client installation option installs only the libraries required to connect from the portal server to a remote process server. Using the WebSphere Process Server Client, you can take advantage of the standard WebSphere Portal mechanisms to federate a portal and to create clusters.

You can obtain your own copy of WebSphere Process Server or use the version supplied in the WebSphere Portal CD set. The WebSphere Process Server CDs for the supported operating systems that are included in the WebSphere Portal CD set provide process server code that was developed for WebSphere Portal. Regardless of which process server installation source you use, you must obtain and run the latest version of WebSphere Update Installer from IBM Support or from the appropriate CD in the WebSphere Portal CD set. Then you must apply process server interim fixes.

Follow these steps:

1. Obtain and run the latest version of WebSphere Update Installer.
2. Install the supported version of WebSphere Process Server; see the System requirements link below for information.
3. Apply the appropriate WebSphere Process Server interim fixes.

Supplemental software provided by WebSphere Portal

Find the interim fixes in the custom.wbi\maintenance directory of the WebSphere Process Server CD for a given operating system.

Find the latest version of the WebSphere Update Installer from IBM Support or from the WebSphere Application Server Network Deployment Supplements CD for a given operating system.

The WebSphere Portal CD set is labeled according to operating system. Find the CDs containing the supplemental software that you need for business process integration by referring to the operating system codes.

The operating system codes are:
- AIX (32-bit and 64-bit) = A
- HP-UX (32-bit) = H
- HP-UX (64-bit) = HI
- i5/OS = I
- Intel Linux (32-bit and 64-bit) = IL
- PowerPC Linux (32-bit and 64-bit) = PL
- zLinux (64-bit) = ZL
- Solaris (32-bit and 64-bit) = SS
- Solaris (64-bit) = SO
- Windows (32-bit and 64-bit) = W

Note: (UNIX only) After copying the content, set read and execute permissions for users doing the installation.

Parent topic: WebSphere Process Server integration

Next topic: Understanding configuration options for process integration
Related concepts
Installation source location
- Upgrading the Update Installer
- Interim fixes for WebSphere Process Server and WebSphere Enterprise Service Bus V6.1

Related tasks
Adding a BPI-enabled portal server to a managed cell in a single-cell setup

Related Information
System requirements
- Technote 1326771
Understanding configuration options for process integration

Review the configuration scenarios to determine which one meets the needs of your enterprise for SOA process portal support with full-scale business process integration. You can configure the IBM® WebSphere® Portal server, either a stand-alone server or a cluster, and the IBM WebSphere Process Server server, either a stand-alone server or a cluster, in different cells of IBM WebSphere Application Server. Alternatively, you can configure stand-alone or clustered portal servers and process servers in the same cell of IBM WebSphere Application Server.

Configuration scenarios

Consider the following configuration scenarios for business process integration:

- WebSphere Portal and WebSphere Process Server managed in different WebSphere Application Server cells
  - A stand-alone portal server configured to use a stand-alone process server
  - A stand-alone portal server configured to use a process server cluster
  - A portal server cluster configured to use a stand-alone process server
  - A portal server cluster configured to use a process server cluster

- WebSphere Portal and WebSphere Process Server managed in the same WebSphere Application Server cell
  - A single, federated portal server configured to use a single, federated process server
  - A single, federated portal server configured to use a process server cluster
  - A portal server cluster configured to use a single, federated process server
  - A portal server cluster configured to use a process server cluster

Important: When planning to integrate business processes for SOA within your portal environment, be aware of the following restrictions:

- You can only install WebSphere Portal into a Base Application Server profile.
- Installing WebSphere Process Server on top of WebSphere Portal supports only the WebSphere Process Server Client functionality.
- The WebSphere Process Server Client functionality and the complete WebSphere Process Server server are mutually exclusive. Therefore, you enable business process integration by installing the WebSphere Process Server Client and connecting to a remote WebSphere Process Server server.

Deployment choices

Determine how you want to deploy the integrated servers by answering the following questions:

- Should the integrated products, WebSphere Portal and WebSphere Process Server, run as stand-alone servers or in a cluster? The decision to use a cluster is driven mainly by the following factors:
  - The need to support advanced availability
  - The planned load of the portal, which may exceed the capabilities of a single server.

- Should these products run in the same WebSphere Application Server management cell or in different management cells? A single-cell deployment is simpler than a cross-cell deployment because the portal and process server cooperate through the common name space (JNDI) of the single-cell. A cross-cell deployment requires that Lightweight Third-Party Authentication (LTPA) single sign-on (SSO) be enabled in WebSphere Application Server for the processing of Common Secure Interoperability Version 2 (CSlv2). However, single-cell deployment increases the dependencies between the
products. For example, if one of the products requires a maintenance upgrade, the other product is likely to require an upgrade as well. It is important to consider the deployment manager server (DMGR) because it coordinates both products. In addition to these technical factors, various non-technical factors, such as the availability of skilled administrators, will influence your decision.

Consider the following factors when choosing a single-cell setup or a cross-cell setup for business process integration:

- The need for manual remote artifact loading: The remote artifact loader allows the generic portlets to retrieve the process artifacts from the process definition, namely, the XML definitions of the input and output messages required for task processing. In a cross-cell setup, you must manually deploy the process artifacts to the WebSphere Portal cell. If you intend to use the generic portlets and want to manage a simple deployment, choose a single-cell setup.

- One type of server is already in operation: If either WebSphere Portal or WebSphere Process Server is already in operation when the complementary servers are added, it is normally a good practice to use two cells because this will minimize the influence of building up the new servers in the existing environment.

- Availability and skills of the required staff: A cross-cell setup requires an additional management console and creates slightly more complexity for the administrator. If your enterprise is small, the availability of a skilled administrator to manage the integrated deployment might be a factor in choosing a single-cell setup over a cross-cell setup.

The illustration shows the various ways in which business process integration can be configured in combination with clustering. The orange arrows indicate the process of scaling an existing portal installation from a single server into a cluster, while the green arrows indicate the configuration step for business process integration. These two actions can be performed in any order. To build up a clustered portal server that is connected to a process server, choose the sequence that meets your needs:

- Cluster first. This sequence is most often used when WebSphere Portal is already installed and running, before the decision to use WebSphere Process Server is made, because availability or the system load required clustering while process server was not available at that time. Once the new process server is available, install the WebSphere Process Server Client on top of the portal to enable portal-to-process server communication:
  1. Install WebSphere Portal as a stand-alone server.
  2. Cluster the portal.
  3. Add the WebSphere Process Server Client on top of each portal node.

- Connect first. This sequence is preferred when WebSphere Process Server is already installed and running, before WebSphere Portal is installed. This approach allows for early prototypes and testing. After prototyping and testing has been successfully completed, the portal will be clustered to carry the load of a production environment.
  1. Install WebSphere Portal as a stand-alone server.
  2. Add the WebSphere Process Server Client on top of each portal node.
  3. Cluster the portal.

*Figure 1. Remote WebSphere Process Server Support*
Remote WebSphere Process Server Support

Parent topic: WebSphere Process Server integration
Previous topic: Prerequisites for setting up process portals
Next topic: Planning to use process portals

Related concepts
Planning for multiple clusters
Planning to use process portals

Consider how you will use business process integration in your portal environment. After you have configured WebSphere Portal to use WebSphere Process Server and enabled business process integration, the pages to manage business processes (My Processes) and human tasks (My Tasks) are available in the portal. The portlets on these pages give you access to following functions:

- Use **My Processes** to work with process templates, process instances, and task templates. You can start new instances of process templates and analyze existing process instances. You can also terminate and delete existing processes.
- Use **My Tasks** to view and manage the tasks assigned to you. View the details of a task, claim a task, and open a claimed task to work on it.

Mappings between the tasks on the process server and task pages on the portal server allow you and other administrators to assign tasks to pages other than the default Task page provided by WebSphere Portal.

When business process integration is enabled and running in your portal deployment, users can work with specific tasks by using the generic portlets provided by WebSphere Portal or custom portlets that have been developed by your integration developers or a third-party developer. During the development phase of your business processes, consider the following questions:

1. Which portlets will provide the best user interface to the business process applications: custom-made portlets or the generic portlets provided by WebSphere Portal? Consider using the portal-provided generic portlets during the initial development of business processes to get prototypes up and running quickly. After the business process applications becomes stable, you can develop custom portlets to refine the user interfaces of the process applications.
2. What is the most effective mapping between the tasks and task pages?
   A. How will we use the default task page?
   B. Will we need to define special-purpose pages dedicated to specific tasks?
   C. What is the initial set of dedicated task pages?
3. What additional portlets will users need on special-purpose task pages to give them the information they need to complete their tasks? Remember to consider portlets that will provide complementary information and that can be placed on the same task page. Avoid forcing users to navigate to other pages.

For complete information about deploying WebSphere Process Server, refer to the IBM® WebSphere® Process Server Version 6.1 Information Center. This information center provides detailed planning information, including planning scenarios. In the section on installing and configuring WebSphere Process Server, you can find information about configuring coexistence with other IBM WebSphere products.

**Parent topic:** WebSphere Process Server integration

**Previous topic:** Understanding configuration options for process integration
Release notes

Known issues and problems are centrally available on the support page. Links into the support knowledge base are integrated throughout the information center to make sure you have the most current information. Before you start the installation process, check the IBM Support site for the most current information about known limitations or issues. Use the following dynamic queries to find late breaking information about this release.

The following links will return the most current list of technotes for the identified area. If a technote has not been published for a given topic area, the link will not return any technotes and you will be instructed to refine your search.

- **Technotes for installation and configuration issues**

- **Technotes for database connectivity issues**

- **All technotes for this release**

You can also search the Support site directly from the information center. See the *Search the IBM support Web site for a solution* topic.

**Parent topic:** Planning for WebSphere Portal

**Related information**

*Search the IBM support Web site for a solution*
Installing WebSphere Portal

IBM® WebSphere® Portal provides flexible deployment options ranging from proof-of-concept where you can examine and test functionality to a highly available and scalable production environment.

Attention: After installing WebSphere Portal and if you plan to use Mashup integration, you will need to run the deploy-portal-mashup-ui task listed in the Integrating > Integrating mashups > Configuring your portal and mashups > Enabling mashup integration in your portal (mandatory) topic.

Select the type of setup you need from the list of options. Follow the scenario to install and configure WebSphere Portal.

- **Installation methods, options, and sources**
  There are multiple installation methods (silent, graphical user interface, and console), options (full or administrator), and sources (CD or CD images).

- **Setting up a single server**
  This scenario is suited to people investigating WebSphere Portal features and capabilities (proof-on-concept), demonstrating the features to interested third-parties, or developers setting up a development environment. This installation path is intended to provide a simple configuration with all the needed components on one server. The instructions for this single-server configuration are Windows specific. The configuration uses a local DB2 database, so you will transfer data from the default database to DB2. However, for test, proof-of-concept environments, and development environments, you can use the out-of-box database. Security is configured as the default federated repository with a built-in file repository.

- **Setting up a stand-alone production server**
  Set up a stand-alone production environment when you do not need a robust clustered environment. A stand-alone server deployment is also useful to determine and validate the needs of your deployment. It enables you to examine and test the functions and features to decide how to accomplish business goals.

- **Setting up a cluster**
  Clusters enable you to scale your WebSphere Portal configuration. Clusters also enable enterprise applications to be highly available because requests are automatically routed to the running servers in the event of a failure. There are numerous cluster configuration, such as horizontal, vertical, multiple, and dynamic.

- **Backing up and restoring profiles**
  To prevent loss of data, you can backup and restore IBM WebSphere Portal profiles running scripts from a QShell session or scheduling scripts as part of a comprehensive backup and restore strategy. The scripts are found in the default directory /QIBM/ProdData/PortalExpress/V61/Tools.

- **Manual database creation and setup**
  For most databases, tasks are provided to create the database (create-database), setup the database users (setup-database), and transfer the default data to the database (database-transfer). However, if you want to create databases manually and transfer the data manually, you can use the alternate configuration instructions provided in this section.

- **About ReleaseBuilder**
  To generate or stage follow-on releases of IBM WebSphere Portal portals, configurations, and artifacts need to be moved between systems. ReleaseBuilder enables management of release configurations independent of user configurations.
Release configuration data are exported to XML files that can be imported using the XML configuration interface (XmlAccess). Using ReleaseBuilder it is possible to stage release configurations between two portals. This allows you to track which configuration entities were removed, added, or changed compared to the previous release generated from a given portal and to apply these differential updates to another portal. Detecting the differences between one configuration and another of the same portal server creates differential updates. A third configuration or “diff”, generated by ReleaseBuilder, represents the changes made between the two configurations. The diff can be used to apply not only addition and update modifications but also deletions to the target server. This allows two portal servers, for example, a staging server and a production server, to remain in synch. ReleaseBuilder is designed to eliminate the need to generate complete XmlAccess exports to move a partial configuration or to manually create XML response files to export a partial configuration. ReleaseBuilder also helps to prevent the problem of configuration bloat on the target server.

- **Setting up a Personalization server on WebSphere Application Server**
  The WebSphere Portal product installs and configures Personalization software by default. However, you can choose to install Personalization to an instance of WebSphere Application Server. When Personalization is installed in this way, it is referred to as the Personalization Server. The Personalization Server enables you to deploy applications that have been developed with IBM WebSphere Application Server, serve personalized content to users, and record site metrics, without requiring that WebSphere Portal also be present on the server machine. While business rules may be run from any WebSphere Application Server installation, a full WebSphere Portal installation is still required as a workspace for authoring rules and campaigns.

- **Uninstalling WebSphere Portal**
  Uninstalling WebSphere Portal is a multiple step process and the method you use is dependent upon your configuration. Removing WebSphere Portal in a single-server configuration is different from removing WebSphere Portal from a cluster. Manual uninstallation instruction are provided for a single-server configuration in case of an error situation.

- **Applying fixes**
  Periodically fix packs are released to integrate product code fixes. Between fix pack releases, interim fixes may be recommended or required to ensure product reliability and stability. The Recommended fixes page provides links to fix pack and interim fix downloads, information about what is recommended and what is required, and links to recommended related product fixes. To find the most current service update information, see the Recommended fixes link.

**Related Information**
- Deployment scenarios
Installation methods, options, and sources

There are multiple installation methods (silent, graphical user interface, and console), options (full or administrator), and sources (CD or CD images).

1. **Installation options**
   
   There are two types of installation: full and administration.

2. **Installation methods**
   
   Select the appropriate installation method for your environment. You can use either a graphical interface program, console mode, or a response file for a silent installation.

3. **Installation source location**
   
   Before installing IBM WebSphere Portal, choose the installation source location and method that best fit your environment. For example, you can install from the CD-ROM or from a file system and you can install using the graphical interface program or perform a silent installation.

**Parent topic:** Installing WebSphere Portal
Installation options
There are two types of installation: full and administration. Choose one of the following installation options:

**Note:** The bit architecture of the IBM® WebSphere® Portal installation program will follow the bit architecture of the Operating System or if WebSphere Portal is being installed into an existing supported IBM WebSphere Application Server, the installation will follow the bit architecture of the WebSphere Application Server. For example, if you are installing onto a 64-bit Linux, WebSphere Portal will install as a 64-bit version. You can force a 32-bit application installation onto a 64-bit operating system; see Installation methods for information.

- **Administration**
  - The administration option includes the administration portlets and welcome pages. Select this option if you do not need all of the out-of-box features. You can add composite applications or common mail portlets later if needed. For example if you plan to design a custom portal environment and only need composite applications, select the Administration options. After you complete the installation and setup, run the appropriate task to install composite applications.

- **Full**
  - The full option includes all of the out-of-box portlets and administration portlets. Select this option if you plan to use most of the out-of-box features.

WebSphere Portal provides a portal light mode which can improve portal start up time and reduce memory consumption in production environments; see Configuring WebSphere Portal > Configuring portal behavior > Using portal light mode for information.

Parent topic: Installation methods, options, and sources
Next topic: Installation methods

Related tasks
Using portal light mode
Adding features to an administration installation
Installation methods
Select the appropriate installation method for your environment. You can use either a graphical interface program, console mode, or a response file for a silent installation.

- **Graphical user interface**
  - The graphical user interface program gathers essential information, such as host name and node, and then performs the installation. Some operating systems do not support a graphical user interface.

- **Console mode**
  - The console mode is a textual representation of the graphical user interface program. Enter a number that corresponds to your selection and then press Enter to proceed through the installation.

- **Silent installation**
  - The silent installation uses a response file to supply installation options without user interaction. Use this option when installing or uninstalling IBM® WebSphere® Portal on multiple machines with a similar setup. To configure the installation, change the options in the response file before issuing the installation task.

**Installation notes:** Read the following information before installing:

- Applies to all installation methods:
  - Before you begin installing or uninstalling, disable or turn off any screen saver software that is running on the machine. This software can affect the ability of the machine to connect to the network and sometimes interfere with the operation of the installation program.
  - The installation program verifies the operating system and its prerequisites, available disk space, and any required software prerequisites prior to installation.
  - You cannot install two instances of the server at the same time, even if you are installing to different directories. You must install each server completely before installing the next one.
  - The installation directory that you specify must NOT contain any files.
  - The installation directory that you specify must NOT contain the following characters: 
    
    - `~`
    - `!`
    - `@`
    - `#`
    - `$`
    - `%`
    - `^`
    - `&`
    - `*`
    - `( single quote )`
    - `- `-`
    - `+`
    - `{`
    - `}`
    - `|`
Applies to both the graphical user interface and the console mode methods: If the host name displays incorrectly, for example `serve1.server1.rtp.raleigh.ibm.com`, you must correct the information; for example, change `serve1.server1.rtp.raleigh.ibm.com` to `server1.rtp.raleigh.ibm.com`. **Note:** This behavior is only found when the installation is run on a Windows server where an instance of Microsoft Exchange server is already installed.

- Applies only when using the graphical user interface:
  - When a dialog is displayed and you must enter text, you must use either your pointing device or press the **TAB** key to move between the entry fields.

### 32 bit installation on 64 bit operating system

Choose one of the following methods to install a 32 bit installation on a 64 bit operating system: **Table 1. 32 bit installation on a 64 bit operating system methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual installation</td>
<td>Manually install the 32-bit IBM WebSphere Application Server product. Then install WebSphere Portal into the existing WebSphere Application Server installation. <strong>Important:</strong> If you have a Solaris Version 9 operating system and you choose this manual method, you MUST manually install the 32-bit WebSphere Application Server product to prevent the installer from automatically selecting a 64-bit environment.</td>
</tr>
<tr>
<td>Command line installation</td>
<td>Enter the `install.bat</td>
</tr>
</tbody>
</table>
Related tasks
Installing WebSphere Portal on AIX on the primary node
Installing WebSphere Portal on AIX on the additional nodes
Installing WebSphere Portal on AIX
Installing WebSphere Portal on HP-UX on the primary node
Installing WebSphere Portal on HP-UX on the additional nodes
Installing WebSphere Portal on HP-UX
Installing WebSphere Portal on IBM i5/OS on the primary node
Installing WebSphere Portal on i5/OS on the additional nodes
Installing WebSphere Portal on IBM i5/OS
Installing WebSphere Portal on Linux on the primary node
Installing WebSphere Portal on Linux on the additional nodes
Installing WebSphere Portal on Linux
Installing WebSphere Portal on Solaris on the primary node
Installing WebSphere Portal on Solaris on the additional nodes
Installing WebSphere Portal on Solaris
Installing WebSphere Portal on Windows on the primary node
Installing WebSphere Portal on Windows on the additional nodes
Installing WebSphere Portal on Windows
Installation source location
Before installing IBM® WebSphere® Portal, choose the installation source location and method that best fit your environment. For example, you can install from the CD-ROM or from a file system and you can install using the graphical interface program or perform a silent installation.

**Note:** You must download, extract, and install the disc images on the same platform that is supported by that image. For example, if you are preparing to install on a Linux platform, you must download, extract, and install the Linux images on a Linux system.

You should take your local environment into consideration when choosing an installation source because installation time can vary. Choose one of the following source locations:

- **Install from the CD-ROM**
  - This option has the following benefits and is best if performing a limited number of installations:
    - No prerequisite steps required before installation
    - Provides an installation path if a file server is not available or is slower than the CD-ROM
  - This install option covers up to 6 CDs; you will be prompted to insert each CD as it is needed.

  **HP-UX note:** When you are prompted to change discs during installation, the eject button on your CD-ROM drive might not be responsive. If this occurs, you must unmount the drive before you can change discs. For example, if the /cdrom directory is the mount point for your drive, you can perform the following steps to change discs:
  1. Enter the `pfs_umount /cdrom` command to unmount the drive.
  2. Change the disc in the CD-ROM drive.
  3. Enter the `pfs_mount /dev/dsk/c3t2d0 /cdrom` command to mount the drive (for example, drive /dev/dsk/c3t2d0).

  **Linux note:** For security reasons, some Linux versions, such as Red hat Enterprise Linux Version 5, prevent programs from executing from automounted CDs. If this is the case on your system, you will be unable to run the installer from CD. To fix this issue, add the `/dev/hdc /media/auto
  pamconsole,fscontext-system_u:object_r:removable_t,exec,noauto,managed 0 0` line to the end of the `/etc/fstab` file to update the system's file system table.

- **Copy CD content to a local machine**
  - This option has the following benefits and is best if repeatedly installing the same product on the same machine:
    - Not dependent on network or CD-ROM speed
    - The installation program finds the CD it needs
  - Perform the following steps to copy CD content to a local machine:
    1. Create a directory for the product; for example, `/wp/version_number`
    2. Copy the Setup CD plus CD 1 through 5 into its own directory; for example:

      **Note:** If your operating system is the 64-bit only zLinux (LZ) or Solaris (SO) system, there is no CD 1, because that CD is for the 32-bit IBM WebSphere Application Server image.

      - `/wp/version_number/OS_code-Setup`
      - `/wp/version_number/OS_code-1`
      - `/wp/version_number/OS_code-2`
      - `/wp/version_number/OS_code-3`
      - `/wp/version_number/OS_code-4`
The operating system codes are:
- AIX (32-bit and 64-bit) = A
- HP-UX (32-bit) = H
- HP-UX (64-bit) = HI
- i5/OS = I
- Intel Linux (32-bit and 64-bit) = IL
- PowerPC Linux (32-bit and 64-bit) = PL
- zLinux (64-bit) = ZL
- Solaris (32-bit and 64-bit) = SS
- Solaris (64-bit) = SO
- Windows (32-bit and 64-bit) = W

**Note:** (UNIX only) After copying the content, set read and execute permissions for users doing the installation.

---

**Copy CD content to a file server**

- This option has the following benefits and is best if installing on multiple machines:
  - Installing from a network drive may be faster than from a CD-ROM drive; review your network and hardware options to determine the best choice
  - The installation program finds the CD it needs

Perform the following steps to copy CD content to a file server:

**Note:** When installing WebSphere Portal on a Windows client using this method, you must mount the file server to a drive letter. The install will not work from a UNC resource (/\servername/mountpoint), which does not have an assigned drive letter.

1. Create a directory for the product; for example, `/wp version_number`
2. Copy the Setup CD plus CD 1 through 5 into its own directory; for example: **Note:** If your operating system is the 64-bit only zLinux (LZ) or Solaris (SO) system, there is no CD 1, because that CD is for the 32-bit IBM WebSphere Application Server image.
   - `/wp version_number OS_code-Setup`
   - `/wp version_number OS_code-1`
   - `/wp version_number OS_code-2`
   - `/wp version_number OS_code-3`
   - `/wp version_number OS_code-4`
   - `/wp version_number OS_code-5`

The operating system codes are:
- AIX (32-bit and 64-bit) = A
- HP-UX (32-bit) = H
- HP-UX (64-bit) = HI
- i5/OS = I
- Intel Linux (32-bit and 64-bit) = IL
- PowerPC Linux (32-bit and 64-bit) = PL
- zLinux (64-bit) = ZL
- Solaris (32-bit and 64-bit) = SS
- Solaris (64-bit) = SO
- Windows (32-bit and 64-bit) = W

**Note:** (UNIX only) After copying the content, set read and execute permissions for users doing the installation.
Setting up a single server

This scenario is suited to people investigating WebSphere Portal features and capabilities (proof-on-concept), demonstrating the features to interested third-parties, or developers setting up a development environment. This installation path is intended to provide a simple configuration with all the needed components on one server. The instructions for this single-server configuration are Windows specific. The configuration uses a local DB2 database, so you will transfer data from the default database to DB2. However, for test, proof-of-concept environments, and development environments, you can use the out-of-box database. Security is configured as the default federated repository with a built-in file repository.

- **Prerequisites**
  - Technotes for installation and configuration issues

The topology below illustrates the single-server configuration that will result from following the instructions in this scenario.

Even for a single-server configuration, you should tune the servers in your portal environment in order to achieve better performance. The Base Portal Tuning scenarios in the WebSphere Portal and Lotus Web Content Management Performance Tuning Guide provides information about tuning the WebSphere Application Server, WebSphere Portal services, databases, directory servers and more.

1. **Preparing your Windows operating system**
   View information on setting up your operating system for IBM WebSphere Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

2. **Installing WebSphere Portal on Windows**
   IBM WebSphere Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

3. **Changing to developer mode**
   Developer mode allows you to develop portals and portlets within an environment that has improved start up performance.

4. **Installing DB2 on the same server as WebSphere Portal**
   View information on installing DB2 for use with WebSphere Portal.
5. **Modifying database properties**
   Learn how to modify the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

6. **Setting up databases**
   This section provides information on using ConfigEngine tasks to create databases and users.

7. **Configuring JCR collation support**
   View the steps to set up JCR collation to work with your IBM DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

8. **Optional: Assigning custom table spaces**
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

9. **Configuring WebSphere Portal to use DB2**
   View information on manually transferring data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

10. **Configuring DB2 for large file handling in Web Content Management**
    If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

11. **Verifying databases**
    After you configure IBM WebSphere Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the `SystemOut.log` and `SystemErr.log` files.

**Parent topic:** Installing WebSphere Portal

**Related concepts**

**Understanding cross-cell deployment scenarios**
Preparing your Windows operating system

View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

Perform the following steps to prepare your operating system:

1. Check that the system logon user ID you will use during installation has the following permissions and rights:
   - The user ID must already exist prior to installation.
   - The user ID must belong to the Administrators group.

2. Perform the following steps to determine if a user account is a member of the Administrators group:
   A. Click Start > Programs > Administrative Tools > Computer Management.
   B. Expand Local Users and Groups and select Groups.
   C. Open the Administrators group to see what members belong to it.
   D. Add the user to the Administrators group if necessary.

3. Consider the following recommendations when installing to avoid excessively long path names:
   - If you exceed the 259 maximum character length, you may receive one of the following error messages during configuration or in the wpinstalllog.txt file:
     - The input line is too long.
     - The syntax of the command is incorrect.
     - The filename is too long.
   A. Use a short installation path. For example, use C:\WebSphere instead of C:\Program Files\IBM\WebSphere.
   B. Specify node names; do not use names longer than 5 characters. For example, you might use node1 instead of longnodename01.
   C. Name WAR files with less than 21 characters. If necessary, modify the file name before installing.

Parent topic: Setting up a single server
Next topic: Installing WebSphere Portal on Windows

---

| 194 |
Installing WebSphere Portal on Windows in a single server environment

IBM® WebSphere® Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

Review the Supported hardware and software requirements for your operating system and ensure that all hardware and software matches the minimum product level before installing WebSphere Portal. Review Installation methods, options, and sources before installing to determine your installation source location and installation method. **Note:** If you plan to use IBM WebSphere Application Server Version 7.0, you must purchase and install it before installing WebSphere Portal on the existing WebSphere Application Server Version 7.0 instance.

**Prerequisites**

- Preparing your Windows operating system

Perform the following steps to install WebSphere Portal:

1. Type `ping yourserver.yourcompany.com` on a command line to verify that your fully qualified host name is properly configured.
2. Type `ping localhost` on a command line to verify that your network is properly configured.
3. If you are installing with an existing WebSphere Application Server instance, ensure it is installed at the supported level and has the following features installed:
   - **Restriction:** Installing WebSphere Portal Version 6.1.0.1 or higher on an existing WebSphere Application Server Version 7.0 instance where WebSphere Portal Version 7.0 was installed is unsupported.
   - **Important:** Besides ensuring that the existing WebSphere Application Server instance is at the supported level, you will also need to ensure that Apache Derby is installed at the supported level before installing WebSphere Portal. See WebSphere Portal detailed system requirements for supported levels.

- Application Server
- Administration
  - Scripted Administration
  - Administrative Console
- Ant and Deployment Tools
  - Deploy Tool
  - Ant Utilities

4. If you are installing on a server with a firewall, antivirus, and/or desktop search engine enabled, disable them before installing. If you do not disable them and the installation program detects them, a warning message displays during the installation.

5. Choose one of the following installation commands based on the installation method you want to use to install WebSphere Portal; see Installation Methods for more information:

   **Advance installation parameters:** To customize your installation, you can add various parameters to your installation commands; for example, you can specify your port information. See Advanced installation parameters for information.

   **Restriction:** When defining your installation path, the ONLY supported characters are ASCII letters [A-Z and a-z], numbers [0-9], slashes [/ or \, depending on your operating system], and the dash [-].
Table 1. Installation task options

<table>
<thead>
<tr>
<th>Installation type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphical user Interface</td>
<td>install.bat</td>
</tr>
<tr>
<td>Console mode</td>
<td>install.bat -console from the PortalExpress subdirectory</td>
</tr>
<tr>
<td>Silent install</td>
<td>install.bat -options &quot;path_to_file\response_filename&quot;, where path_to_file is the full path to the response file and response_filename is the name of the file. A sample install response file (installresponse.txt) and a sample uninstall response file (uninstallresponse.txt) are located in the setup CD root directory. Important: Do not place the response file in a path that contains a space and do not put a space in the file name.</td>
</tr>
</tbody>
</table>

**Note:** If the installation program does not detect a WebSphere Application Server instance that you know exists, exit the installation program and pass the WebSphere Application Server instance location using the command line; for example, `install.bat -W was.undetectedWas="/my/WAS/location"`. **Note:** If the GUI or console mode installation program fails to detect ports for either WebSphere Application Server or WebSphere Portal, a warning message displays and the installer offers another chance to enter the values. If the silent installation fails to detect ports for either WebSphere Application Server or WebSphere Portal, the installer will exit.

6. Verify your installation was successful; access WebSphere Portal using the `http://yourserver:yourport/wps/portal` format. Run the ConfigEngine.bat list-server-ports -DWasPassword=password task from the wp_profile_root \ConfigEngine directory to generate the wp_profile_root\ConfigEngine\log\wp_PortMatrix.txt file that lists the WebSphere Portal ports for your installation.

7. Optional: After the WebSphere Portal installation completes successfully, run the ConfigEngine.bat configure-express -DPortalAdminPwd=password -DWasPassword=password task, from the wp_profile_root\ConfigEngine directory, if you want to enable the sample IBM Lotus Web Content Management content, such as internet and intranet sample sites.

**Restriction:** Run the configure-express task before configuring your database, user registry, context root, security, etc. If you ran any tasks other than the install task, do not run this task.

The sample content includes: **Note:** See [http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_ovw.html](http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_ovw.html) for tutorials on how to use the sample content.

- Creates a group called contentAuthors; members of this group are given privileges to create content in the sample Internet and intranet sites. Navigate to the Administration area and then click Access > Users and Groups.
- Creates two new Web Content Management Libraries: "Internet Web Content 6.1.0" and "Intranet Web Content 6.1.0". Navigate to the Administration area and then click Portal Content > Web Content Libraries.
- Adds a portlet filter and applies the filter to various portlets in the sample Internet and intranet sites. You can see the definition of the filter in the WebSphere Application Server Administration console and examining the custom resources under the Environment area.
- Creates two new theme policies: InternetStyle and IntranetStyle. These styles are applied to sample Internet and intranet sites. Navigate to Theme Customizer and then select the style.
- Creates several portlet clones of the Web Content Management rendering portlet. These portlet clones are used on sample Internet and intranet sites.
- Creates two virtual portals with context roots of wps/portal/intranet and wps/portal/internet. These are the sample Internet and intranet sites. Go to [http://yourserver:yourport/wps/portal/internet](http://yourserver:yourport/wps/portal/internet) and
http://yourserver:yourport/wps/portal/intranet to access them.
- Creates several sample credential slots, including "Default slot for E-mail", "Default slot for Feeds", "Default slot for Miscellaneous", "Default slot for Web Clipping", and "Default slot for Web Content Management". Navigate to the Administration area and then click Access > Credential Vault > Manage System Vault Slots.
- Modifies the portlet palette to include some new portlets. Click the Portlets link to see the portlet palette.

8. Optional: If you ran the configure-express task, the owner of the items in the Web content libraries containing the Internet and Intranet Site Template content will be listed as uid=xyzadmin,o=defaultWIMFileBasedRealm. To update the owner information for these items to correspond to your portal administrator ID, complete the following steps:
   A. Edit the wp_profile_root\PortalServer\wcm\shared\app\config\wcmservices\MemberFixerModule.properties file.
   B. Add the following line to the file:
      
      `uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN`
      
      Replace `portal_admin_DN` with the distinguished name of the portal administrator.
   C. Save your changes and close the file.
   D. Run the ConfigEngine.bat action-express-memberfixer -DPortalAdminPwd=password -DWasPassword=password task, located in the wp_profile_root\ConfigEngine directory.

9. Optional: Perform the following steps if you need to change the ports for WebSphere Application Server or WebSphere Portal:

   **Note:** The starting port parameter is required for a successful completion of the modify-ports-by-startport task.

   Once you specify a start port, this port becomes the base for assigning port values. The code increments this value as each port is assigned, which means that the WebSphere Portal ports will be assigned incrementally starting with the port defined with the modify-ports-by-startport task.

   A. Stop the server1 and WebSphere_Portal servers.
   B. Run one of the following commands for each server you need to change:

   C. Restart the server1 and WebSphere_Portal servers.

<table>
<thead>
<tr>
<th>Method</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting port number</td>
<td>ConfigEngine.bat modify-ports-by-startport -DWasPassword=password -DModifyPortsServer=servername -DStartPort=starting port number</td>
</tr>
</tbody>
</table>

**Note:** Sample port files are available on the Setup disc.

The following is an example of the information within a port file although the port values will be different based on your environment:

```
BOOTSTRAP_ADDRESS=10031
SOAP_CONNECTOR_ADDRESS=10033
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=10032
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=10025
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=10036
WC_adminhost=10027
WC_defaulthost=33344
DCS_UNICAST_ADDRESS=10029
WC_adminhost_secure=10027
WC_defaulthost_secure=33344
SIB_ENDPOINT_ADDRESS=10026
SIB_ENDPOINT_SECURE_ADDRESS=10028
ORB_LISTENER_ADDRESS=10034
```
Changing to developer mode

Developer mode allows you to develop portals and portlets within an environment that has improved start up performance. Install IBM® WebSphere® Portal with the Administration option.

- **Prerequisites**
  - Preparing your Windows operating system
  - Installing WebSphere Portal on Windows

Choose one of the following options to change to developer mode immediately after installing WebSphere Portal:

- **Configuring developer mode on Windows**
The developer mode allows you to improve start up performance and to configure IBM WebSphere Portal for development. This file explains how to configure and unconfigure the development mode option. Developer mode is for a development environment only and should not be used in a production environment.

- **Configuring developer mode on AIX**
The developer mode allows you to improve start up performance and to configure IBM WebSphere Portal for development. This file explains how to configure and unconfigure the development mode option. Developer mode is for a development environment only and should not be used in a production environment.

- **Configuring developer mode on HP-UX**
The developer mode allows you to improve start up performance and to configure IBM WebSphere Portal for development. This file explains how to configure and unconfigure the development mode option. Developer mode is for a development environment only and should not be used in a production environment.

- **Configuring developer mode on IBM i5/OS**
The developer mode allows you to improve start up performance and to configure IBM WebSphere Portal for development. This file explains how to configure and unconfigure the development mode option. Developer mode is for a development environment only and should not be used in a production environment.

- **Configuring developer mode on Linux**
The developer mode allows you to improve start up performance and to configure IBM WebSphere Portal for development. This file explains how to configure and unconfigure the development mode option. Developer mode is for a development environment only and should not be used in a production environment.

- **Configuring developer mode on Solaris**
The developer mode allows you to improve start up performance and to configure IBM WebSphere Portal for development. This file explains how to configure and unconfigure the development mode option. Developer mode is for a development environment only and should not be used in a production environment.

**Parent topic:** Setting up a single server

**Previous topic:** Installing WebSphere Portal on Windows

**Next topic:** Installing DB2 on the same server as WebSphere Portal

**Parent topic:** Configuring WebSphere Portal
**Related concepts**
- Configuration task properties
- Configuring WebSphere Portal with the configuration wizard
- Configuring portal behavior
- Configuring the IBM License Metric Tool and IBM Tivoli License Compliance Manager

**Related tasks**
- Changing the portal URI
- Configuring Web Content Management
- Setting up a remote spell checker
- Enabling Document Conversion Services
- Connecting to existing database domains
- Adding features to an administration installation
- Enabling FIPS

**Related information**
- Managing the user registry
- Additional security features
Configuring developer mode on Windows

The developer mode allows you to improve start up performance and to configure IBM® WebSphere® Portal for
development. This file explains how to configure and unconfigure the development mode option. Developer mode is for a
development environment only and should not be used in a production environment.

Install WebSphere Portal with the Administration installation option.

This task modifies the following components:

- **JVM**
  - The JVM will be switched to development mode and the initial heap size will be set to the maximum heap size to reduce the amount of garbage collection during start up.

- **Portlets**
  - Portlets and Web Applications will be activated on first access and not at the start up. Since some of the portlets and applications are required at start up, a white list, which contains the list of applications, will hold the applications still started at start up. **Note:** To add applications to the white list, modify the `profile\PortalServer\config\StartupPerformance\wp.base_ProfileEarAttributesTargetMapExclList.jacl` file.  
  Add a line such as `lappend WarFileNameList App_name`, where `App_name` is the name of the application. Log on to the WebSphere Application Server Console Mode and navigate to Applications > Application Types > WebSphere enterprise applications to get a list of available applications.

Run the `ConfigEngine.bat enable-develop-mode-startup-performance -DWasPassword=password` task, from the `wp_profile_root\ConfigEngine` directory, immediately after installing WebSphere Portal to develop portals and portlets.

Prepare the remote Web server for your developer mode. See the appropriate link below for information.

Run the `ConfigEngine.bat disable-develop-mode-startup-performance -DWasPassword=password` task, from the `wp_profile_root\ConfigEngine` directory, to revert back to a production server. **Note:** You can run the `disable-develop-mode-startup-performance` task when you are done developing your portal and portlets, if the development settings are not adequate for a special development situation, or if a problem found on the production server cannot be recreated on the development server.

**Parent topic:** Changing to developer mode

**Parent topic:** Changing to developer mode

**Related tasks**

Preparing a remote Web server when portal is installed on Windows
Installing DB2 on the same server as WebSphere Portal

View information on installing DB2 for use with WebSphere Portal.

- Prerequisites
  - Preparing your Windows operating system
  - Installing WebSphere Portal on Windows

1. Before installing DB2, log in with a user ID that has administrative authority. This user should have the following specifications:
   - Belong to the local Administrator group
   - Act as part of the operating system
   - Have permissions to create a token object
   - Windows 2003 only: Have permissions to adjust memory quotas for a process
   - Have permissions to replace a process level token

   To edit user rights:
   - For the first two specifications: Click Start > Programs > Administrative Tools > Computer Management > Local Users and Groups.
   - For the last four specifications: Click Start > Programs > Administrative Tools > Local Security Policy. Then, click Local Policies > User Rights Assignment.

2. To install DB2, follow the instructions that are provided with the DB2 documentation to install DB2 and the required fix pack.

Parent topic: Setting up a single server
Previous topic: Changing to developer mode
Next topic: Modifying database properties

Related information
- IBM DB2 Database 9.1 for Linux, UNIX, and Windows Information Center
- IBM DB2 Database Version 8 for Linux, UNIX, and Windows Information Center
Modifying database properties

Learn how to modify the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

- **Prerequisites**
  - Preparing your Windows operating system
  - Installing WebSphere Portal on Windows
  - Installing DB2 on the same server as WebSphere Portal

The `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 `wkplc.properties` file reference
- WebSphere Portal 6.1.5 `wkplc_comp.properties` file reference
- WebSphere Portal 6.1.5 `wkplc_dbtype.properties` file reference

**Working with properties files:**
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:
  - `release.DbName=release`
  - `jcr.DbName=jcrdb`
  - `feedback.DbName=fdbkdb`
  - `likeminds.DbName=lmdb`
  - `community.DbName=commdb`
  - `customization.DbName=custdb`
- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace `dbdomain` with:
  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
- `dbdomain.DbType`
- `dbdomain.DbName`
- `dbdomain.DbUrl`
- `dbdomain.DbSchema`

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If `DbUser`, `DbUrl`, and `DbPassword` are not the same across domains, the value for `DataSourceName` must differ from the `DataSourceName` of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   - `wp_profile_root/ConfigEngine/properties/wkplc.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_comp.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplcDbType.properties`

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In `wkplc_comp.properties`, most properties are repeated for each domain.

2. Use a text editor to open the properties file `wkplc_comp.properties` and modify the values to correspond to your environment. **Note:** The `wkplc_comp.properties` file by default has the properties populated for the Apache Derby database. The source `dbdomain.properties` represent the properties for the source database. These values should not be modified unless you are transferring from a supported database other than Apache Derby.
   A. For `dbdomain.DbType`, type `db2`.
   B. For `dbdomain.DbName`, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the `dbdomain.DbUrl` property.
   C. For `dbdomain.DbName`, type the name of the WebSphere Portal domain database. DB2 database names cannot exceed eight (8) characters. **Note:** This value is also the database element in the `dbdomain.DbUrl` property.
   D. For `dbdomain.DbSchema`, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   E. For `dbdomain.DataSourceName`, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - `releaseDS`
      - `communityDS`
      - `customizationDS`
      - `jcrDS`
      - `lmdbDS`
      - `feedback`
   F. For `dbdomain.DbUrl`, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:** The database element of this value should match the value of DbName.
   G. For `dbdomain.DbUser`, type the user ID for the database administrator.
   H. For `dbdomain.DbPassword`, type the password for the database administrator.
   I. Optional: For `dbdomain.DbRuntimeUser`, type the user ID of the database user that should be used by WebSphere Portal to connect to the database at runtime.
   J. If `dbdomain.DbRuntimeUser` is specified, you must set `dbdomain.DbRuntimePassword` to be the password of the runtime database user.
   K. For `dbdomain.DBA.DbUser`, type the database administrator user ID for privileged access operations during creation of the database.
L. For *dbdomain*.DBA.DbPassword, type the database administrator password for privileged access operations during creation of the database.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

   A. For *source.domain*.DbType, type of the database you are currently configured to use. The value for *source.domain* .DbType is Derby by default.

   B. For *source.domain*.DbName, type the name of the database domain you are currently using.

   C. For *source.domain*.DbSchema, type current schema identifier for objects within the database for this domain.

   D. For *source.domain*.DataSourceName, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

   E. For *source.domain*.DbUrl, type the url currently used to access your database.

   F. For *source.domain*.DbUser, type the name of the user accessing this database.

   G. For *source.domain*.DbPassword, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

   A. For db2.DbDriver, type the name of the JDBC driver class.

   B. For db2.DbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For db2.JdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

   A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Setting up a single server  
**Previous topic:** Installing DB2 on the same server as WebSphere Portal  
**Next topic:** Setting up databases
Setting up databases

This section provides information on using ConfigEngine tasks to create databases and users.

Before you begin, ensure that the following prerequisites are met:

- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

- **Prerequisites**
  - Preparing your Windows operating system
  - Installing WebSphere Portal on Windows
  - Installing DB2 on the same server as WebSphere Portal
  - Modifying database properties

The following steps are the same for root and non-root users, except that the create-database task cannot be run by a non-root user.

1. Change to the directory `wp_profile_root\ConfigEngine`
2. To create the databases, type the following command:
   ```cmd
   ConfigEngine.bat create-database -DWasPassword=password
   ```
3. Check the services file on the DB2 server system. If it does not specify DB2 connection and interrupt service ports, specify the ports for your operating system.
   A. Use a text editor to open the file `%SYSTEMROOT%\system32\drivers\etc\services`.
   B. Add the text `db2c_db2 50000/tcp`, where `db2` is the default instance.
   **Note:** Ensure the port number used is not already in use. If 50000 is already in use, select a different port number.
4. To create the database users, type the following command:
   ```cmd
   ConfigEngine.bat setup-database -DWasPassword=password
   ```
   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:
   `'<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql'`.

Parent topic: Setting up a single server
Previous topic: Modifying database properties
Next topic: Configuring JCR collation support

Related tasks
Creating local databases on Windows
Configuring JCR collation support

View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

**Prerequisites**
- Preparing your Windows operating system
- Installing WebSphere Portal on Windows
- Installing DB2 on the same server as WebSphere Portal
- Modifying database properties
- Setting up databases

1. Stop the WebSphere Portal server:
   ```
   stopServer.bat WebSphere_Portal -username admin_userid -password admin_password
   ```

2. If using a remote DB2 server, copy the following files from the WebSphere Portal server to a temporary directory on the remote DB2 server:
   ```
   PortalServer_root/jcr/prereq.jcr/config/collation.jar
   PortalServer_root/jcr/prereq.jcr/config/registerCollationUDFTemplate.sql
   ```

3. Set up collation on the database where the JCR domain is located. Log in to the database machine with a userid that is authorized and configured to use the appropriate DB2 instance. For example, a common userid is `db2inst1`.
   A. Change to the directory `db2home/function`.
   B. Execute the command:
      ```
      Remote DB2: db2home/java/jdk/bin/jar -xvf temporary location/collation.jar
      icm/CollationUDF.class
      ```
      ```
      Local DB2: db2home/java/jdk/bin/jar -xvf PortalServer_root/jcr/prereq.jcr/config/collation.jar
      icm/CollationUDF.class
      ```
   C. Change to the directory `wp_profile_root\PortalServer\jcr\config`. If using a remote DB2, change to the temporary directory where you copied the files in a previous step.
   D. Open the file `registerCollationUDFTemplate.sql` and change all `SCHEMA` references to the JCR schema; for example, `JCR`. The value set for `SCHEMA` should match the value set for the jcr.DbSchema property.
   E. Connect to the JCR database by running `db2 connect to <jcrdb> user <userid> using <password>`. Execute the script by running the command:
      ```
      Remote DB2: db2 -tvf temporary location/registerCollationUDFTemplate.sql
      ```
      ```
      Local DB2: db2 -tvf wp_profile_root\PortalServer\jcr\config/registerCollationUDFTemplate.sql
      ```
   F. Disconnect from the JCR database.
   G. Restart the DB2 instance.
   H. Execute the following command in the previous substep. If the command completes successfully, the UDF is registered correctly as follows: `values schema.sortkeyj('abc','en')`, where `schema` is the schema used in the previous substep.

4. Verify that the UDF is registered properly.
   A. Log in as the `db2instanceID`. Open a DB2 terminal window, and type `db2`. From the command line, type the command `connect to JCRDB user userid using password`. You must connect to the JCR database using the userid and password of the WebSphere Portal JCR data source.
   B. Execute the following command in the previous substep. If the command completes successfully, the UDF is registered correctly as follows: `values schema.sortkeyj('abc','en')`, where `schema` is the schema used in the previous substep.
5. Edit the icm.properties file, located in \wp_profile_root\PortalServer\jcr\lib\com\ibm\icm directory. Add the following section to the end of the file:

```
# Enable/Disable collation support for all DB2 platforms
jcr.query.collation.db2.enabled = true

# Database specific collation mappings
# These mappings apply map a Java locale name into a collation name
# supported by the underlying database.
# Example mappings for DB2 platform

# English
jcr.query.collation.en = en

# Swedish
jcr.query.collation.sv = sv

jcr.query.collation.zh = zh
jcr.query.collation.de = de
jcr.query.collation.da = da
jcr.query.collation.hu = hu
jcr.query.collation.jp = jp
```

6. Enter the following command to start the WebSphere Portal server:

```
startServer.bat WebSphere_Portal
```
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:

- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties` file.
- The page size of table spaces used by WebSphere Portal must be 8192 bytes.
- For details on creating table spaces refer to the documentation for the database.

**Prerequisites**

- Preparing your Windows operating system
- Installing WebSphere Portal on Windows
- Installing DB2 on the same server as WebSphere Portal
- Modifying database properties
- Setting up databases
- Configuring JCR collation support

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties` that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`

   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds

3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword `IN` and a space. For example: `community.COMP_INST.tablespace=IN COMM8KSPACE`

   Repeat this step for each domain that you are transferring.
4. Save and close `dbdomain.space_mapping.properties`.
5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
- **Windows**: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`  
- **UNIX**: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`  
- **i5/OS**: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
Configuring WebSphere Portal to use DB2

View information on manually transferring data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

- Prerequisites
  - Preparing your Windows operating system
  - Installing WebSphere Portal on Windows
  - Installing DB2 on the same server as WebSphere Portal
  - Modifying database properties
  - Setting up databases
  - Configuring JCR collation support

Tips:

- To run these tasks as a non-root user, you must first run the task chown -R non-root_userWebSphereDir.

- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

- Be sure that DB2 is started by checking the service. If attempts to restart result in a logon failure message, then go to the DB2 properties and reenter the password.

Steps for transferring data to another supported database

1. If you are running a type 2 connection, before transferring data edit the db2cli.ini file. Failure to follow these steps will cause the database transfer to hang at the task action-process-constraints.
   A. Locate the file C:\Program Files\IBM\SQLLIB\db2cli.ini.
   B. Add the following to the end of the file. Leave an empty line after ReturnAliases=0.[COMMON]

   DYNAMIC=1
   ReturnAliases=0

2. Open a command prompt and change to the directory wp_profile_root\ConfigEngine.

3. Enter the following commands to validate configuration properties. ConfigEngine.bat validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds-DWasPassword=password
   ConfigEngine.bat validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds-DWasPassword=password
4. From the same command prompt as the previous steps, change to the directory `wp_profile_root\bin`.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

```plaintext
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td><code>stopServer.bat server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td><code>stopServer.bat WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>
```

6. Transfer the database:

   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory `wp_profile_root\ConfigEngine`.

   B. Enter the following command:

   ```plaintext
   ConfigEngine.bat database-transfer -DWasPassword=password
   ```

   **Note:**
   - To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:
     ```plaintext
     ConfigEngine.bat database-transfer -DTransferDomainList=jcr -DWasPassword=password
     ```
   - If you have been storing data in IBM® Derby for a long time, database transfer could fail with OutOfMemory exceptions. If database transfer fails, add the following property to the command in this step:
     ```plaintext
     ConfigEngine.bat database-transfer -DDbtJavaMaxMemory=1536M -DWasPassword=password
     ```

   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

7. After transferring the database tables, perform a `reorg` check to improve performance. Perform this step for each database alias in the property file.

   A. Connect to a database with the following command:

   ```plaintext
   db2 connect to database_alias user db2admin_userid using password
   ```

   **Note:** Additional options might be required if additional security has been installed. Refer to DB2 Universal Database commands by example for links to the command reference.

   B. After it is connected, run the following command from the DB2 prompt:

   ```plaintext
   db2 reorgchk update statistics on table all > xyz.out
   ```

   C. Look in the `reorg` column for entries marked with a * (star or asterisk) in the file `xyz.out`. For each line with a *, note the `tablename` and run the following command for each `tablename`:

   ```plaintext
   db2 reorg table tablename
db2 terminate
db2rbind database_name -l db2rbind.out -u db2_admin -p password
   ```

   **Note:** The output file `db2rbind.out` is only created when there is an error for the `db2rbind` command.

8. Change to the directory `wp_profile_root\bin`.

9. Enter the following command to start the WebSphere Portal server:

   ```plaintext
   startServer.bat WebSphere_Portal
   ```

**Parent topic:** Setting up a single server

**Previous topic:** Assigning custom table spaces

**Next topic:** Configuring DB2 for large file handling in Web Content Management
Configuring DB2 for large file handling in Web Content Management

If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

**Prerequisites**
- Preparing your Windows operating system
- Installing WebSphere Portal on Windows
- Installing DB2 on the same server as WebSphere Portal
- Modifying database properties
- Setting up databases
- Configuring JCR collation support
- Configuring WebSphere Portal to use DB2

**Note:** You only need to perform these steps if you are using Web Content Management.

1. Log into the WebSphere Application Server administrative console.
2. Click **Resources > JDBC > Data sources**.
3. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
4. Select the name of the data source that is defined in `wkplc_comp.properties` for the JCR database domain. The default data source is `wpdbDS`.
5. Click **Custom properties**.
6. Ensure that the `fullyMaterializeLobData` property is set to `false`.

**Parent topic:** Setting up a single server

**Previous topic:** Configuring WebSphere Portal to use DB2

**Next topic:** Verifying databases
Verifying databases

After you configure IBM® WebSphere® Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.

- **Prerequisites**
  - Preparing your Windows operating system
  - Installing WebSphere Portal on Windows
  - Installing DB2 on the same server as WebSphere Portal
  - Modifying database properties
  - Setting up databases
  - Configuring JCR collation support
  - Configuring WebSphere Portal to use DB2
  - Technotes for database connectivity issues

You can verify the database connection using IBM WebSphere Application Server or by opening WebSphere Portal in a browser.

- To verify that the WebSphere Portal application server is running by using WebSphere Application Server, complete these steps:

  1. Open the WebSphere Application Server administrative console by entering the following address in a browser:

     http://hostname.example.com:10027/ibm/console

     where `hostname.example.com` is the fully qualified host name of the machine where WebSphere Portal is running and `10027` is the default transport port that is created by WebSphere Application Server.

  2. Log into the administrative console.

  3. Depending on your version of WebSphere Application Server, click the appropriate option:

     - For **WebSphere Application Server Version 6.1**: Click **Resources & JDBC Providers**.
     - For **WebSphere Application Server Version 7.0**: Click **Resources > JDBC > JDBC Providers**

  4. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.

  5. Select the name of the data source that is defined in `wkplc_comp.properties`. The default data source is `wpdbDS`.

  6. Select the name of the JDBC provider that is specified in `wkplc_dbtype.properties`. The default JDBC provider is `wpdbJDBC_dbtype`, where `dbtype` is replaced by the value that matches your environment.

  7. Click **Test Connection** to verify the database connection. If configuration parameters have been changed, you might need to restart WebSphere Application Server for the test to complete.

- To verify that the WebSphere Portal application server is running by opening WebSphere Portal in a browser, enter the following URL in a supported browser:

  http://hostname.example.com:10040/wps/portal

  where `hostname.example.com` is the fully qualified host name of the machine where WebSphere Portal is running and `10040` is the default transport port that is created by WebSphere Application Server.

**Parent topic:** Setting up a single server

**Previous topic:** Configuring DB2 for large file handling in Web Content Management
Setting up a stand-alone production server

Set up a stand-alone production environment when you do not need a robust clustered environment. A stand-alone server deployment is also useful to determine and validate the needs of your deployment. It enables you to examine and test the functions and features to decide how to accomplish business goals.

**- Prerequisites**

- Technotes for installation and configuration issues

The following illustrations depicts the stand-alone server configuration that will result from following the instructions in this scenario.

After completing the installation remember to tune the servers in your portal environment in order to achieve better performance. The Base Portal Tuning scenarios in the WebSphere Portal and Lotus Web Content Management 6.1.x Performance Tuning Guide provides information about tuning the WebSphere Application Server, WebSphere Portal services, databases, directory servers and more. Base Portal Tuning scenarios

- Setting up a stand-alone server on AIX
  After you have prepared the operating system you are ready to install WebSphere Portal.

- Setting up a stand-alone server on HP-UX
  After you have prepared the operating system you are ready to install WebSphere Portal.

- Setting up a stand-alone server on i5/OS
  After you have prepared the operating system you are ready to install WebSphere Portal.

- Setting up a stand-alone server on Linux
  After you have prepared the operating system you are ready to install WebSphere Portal.

- Setting up a stand-alone server on Solaris
  After you have prepared the operating system you are ready to install WebSphere Portal.

- Setting up a stand-alone server on Windows
  After you have prepared the operating system you are ready to install WebSphere Portal.

**Parent topic:** Installing WebSphere Portal

**Related concepts**

Understanding cross-cell deployment scenarios
Design servers
Setting up a stand-alone server on AIX

After you have prepared the operating system you are ready to install WebSphere Portal.

- Prerequisites
  - Technotes for installation and configuration issues

Select the operating system on which you are installing WebSphere Portal.

1. Preparing your AIX operating system
   View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

2. Installing WebSphere Portal on AIX
   IBM WebSphere Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

3. Configuring databases
   Before transferring default data to the production database server, you need to prepare the database server. Database preparation includes creating user IDs and databases. For this installation path, stand-alone production server, remote databases servers are used.

4. Preparing a remote Web server when portal is installed on AIX
   Install and configure the Web server plug-in, provided by IBM WebSphere Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

5. Configuring WebSphere Portal to use a user registry on AIX
   Configure user registry security on IBM WebSphere Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

6. Tune your servers
   Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

Parent topic: Setting up a stand-alone production server
Preparing your AIX operating system

View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

Perform the following steps to prepare your AIX machine:
1. Set the file descriptor limit to 10240; for example, `ulimit -n 10240`.
2. **Web Content Management only:** Use the `ulimit -f` command to set the maximum size of files that can be created to be at least the size of the largest file you would need to upload to the content server. The command `ulimit -f unlimited` removes any limit on file size.
3. Install and configure X server on AIX (for example X-Windows or GNOME) to use the graphical user interface the installation program provides. **Note:** X server is not required if installing with a response file or in console mode.

Parent topic: Setting up a stand-alone server on AIX
Next topic: Installing WebSphere Portal on AIX
Installing WebSphere Portal on AIX

IBM® WebSphere® Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

Review the Supported hardware and software requirements for your operating system and ensure that all hardware and software matches the minimum product level before installing WebSphere Portal. Review Installation methods, options, and sources before installing to determine your installation source location and installation method. Note: If you plan to use IBM WebSphere Application Server Version 7.0, you must purchase and install it before installing WebSphere Portal on the existing WebSphere Application Server Version 7.0 instance.

- Prerequisites
  - Preparing your AIX operating system

Perform the following steps to install WebSphere Portal:

1. Type `ping yourserver.yourcompany.com` on a command line to verify that your fully qualified host name is properly configured.
2. Type `ping localhost` on a command line to verify that your network is properly configured.
3. If you are installing with an existing WebSphere Application Server instance, ensure it is installed at the supported level and has the following features installed:
   - Restriction: Installing WebSphere Portal Version 6.1.0.1 or higher on an existing WebSphere Application Server Version 7.0 instance where WebSphere Portal Version 7.0 was installed is unsupported.
   - Important: Besides ensuring that the existing WebSphere Application Server instance is at the supported level, you will also need to ensure that Apache Derby is installed at the supported level before installing WebSphere Portal. See WebSphere Portal detailed system requirements for supported levels.

- Application Server
- Administration
  - Scripted Administration
  - Administrative Console
- Ant and Deployment Tools
- Deploy Tool
- Ant Utilities

4. Optional: Perform the following steps if you want to install WebSphere Portal using a non-root user:
   - Restriction: Although it is possible to install WebSphere Application Server as a non-root user, there are limitations to this option that can affect WebSphere Portal functionality, which is why you should install WebSphere Application Server as a root user.
   A. Install the current supported version of WebSphere Application Server as a root user.
   B. Open a command prompt and perform the following steps to create a non-root user and to change ownership:

      1. Use the appropriate system commands to create a new group, to create a new user, and to add the new user to the new group.
2. Run the following tasks to change the rights of the non-root user:
   ```bash
   chmod -R g+rwx /usr/IBM
   chgrp -R group_name /usr/IBM
   chmod -R g+wr /tmp
   chgrp -R group_name /tmp
   chmod -R g+wr /var/tmp
   chgrp -R group_name /var/tmp
   ```

C. Launch the installation program per the next step. During the installation, select the existing WebSphere Application Server using the non-root user and set the installation location to a unique location under the path where non-root permissions were granted.

5. Choose one of the following installation commands based on the installation method you want to use to install WebSphere Portal; see Installation Methods for more information:

   **Advance installation parameters:** To customize your installation, you can add various parameters to your installation commands; for example, you can specify your port information. See Advanced installation parameters for information.

   **Restriction:** When defining your installation path, the ONLY supported characters are ASCII letters [A-Z and a-z], numbers [0-9], slashes [/ or \
 depending on your operating system], and the dash [-].

   **Table 1. Installation task options**

<table>
<thead>
<tr>
<th>Installation type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphical user interface</td>
<td>./install.sh</td>
</tr>
<tr>
<td>Console mode</td>
<td>./install.sh -console from the PortalExpress subdirectory</td>
</tr>
<tr>
<td>Silent Install</td>
<td>./install.sh -options &quot;path_to_file/response_filename&quot;, where path_to_file is the full path to the response file and response_filename is the name of the file. A sample install response file (installresponse.txt) and a sample uninstall response file (uninstallresponse.txt) are located in the setup CD root directory. <strong>Important:</strong> Do not place the response file in a path that contains a space and do not put a space in the file name.</td>
</tr>
</tbody>
</table>

**Note:** If the installation program does not detect a WebSphere Application Server instance that you know exists, exit the installation program and pass the WebSphere Application Server instance location using the command line; for example, .\install.sh -W was.undetectedWas="/my/WAS/location".

**Note:** If the GUI or console mode installation program fails to detect ports for either WebSphere Application Server or WebSphere Portal, a warning message displays and the installer offers another chance to enter the values. If the silent installation fails to detect ports for either WebSphere Application Server or WebSphere Portal, the installer will exit.

6. Verify your installation was successful; access WebSphere Portal using the http://yourserver:yourport/wps/portal format. Run the .\ConfigEngine.sh list-server-ports -DWasPassword=password task from the wp_profile_root /ConfigEngine directory to generate the wp_profile_root/ConfigEngine/log/wp_PortMatrix.txt file that lists WebSphere Portal ports for your installation.

7. Optional: After the WebSphere Portal installation completes successfully, run the .\ConfigEngine.sh configure-express -DPortalAdminPwd=password -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, if you want to enable the sample IBM Lotus Web Content Management content, such as internet and intranet sample sites.

   **Restriction:** Run the configure-express task before configuring your database, user registry, context root, security, etc. If you ran any tasks other than the install task, do not run this task.

   **Note:** See http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_ovw.html for more information.
tutorials on how to use the sample content.

The sample content includes:

- Creates a group called contentAuthors; members of this group are given privileges to create content in the sample Internet and intranet sites. Navigate to the **Administration** area and then click **Access > Users and Groups**.

- Creates two new Web Content Management Libraries: "Internet Web Content 6.1.0" and "Intranet Web Content 6.1.0". Navigate to the **Administration** area and then click **Portal Content > Web Content Libraries**.

- Adds a portlet filter and applies the filter to various portlets in the sample Internet and intranet sites. You can see the definition of the filter in the WebSphere Application Server Administration console and examining the custom resources under the **Environment** area.

- Creates two new theme policies: InternetStyle and IntranetStyle. These styles are applied to sample Internet and intranet sites. Navigate to **Theme Customizer** and then select the style.

- Creates several portlet clones of the Web Content Management rendering portlet. These portlet clones are used on sample Internet and intranet sites.


- Creates several sample credential slots, including "Default slot for E-mail", "Default slot for Feeds", "Default slot for Miscellaneous", "Default slot for Web Clipping", and "Default slot for Web Content Management". Navigate to the **Administration** area and then click **Access > Credential Vault > Manage System Vault Slots**.

- Modifies the portlet palette to include some new portlets. Click the **Portlets** link to see the portlet palette.

8. Optional: If you ran the `configure-express` task, the owner of the items in the Web content libraries containing the Internet and Intranet Site Template content will be listed as `uid=xyzadmin,o=defaultWIMFileBasedRealm`. To update the owner information for these items to correspond to your portal administrator ID, complete the following steps:

A. Edit the `wp_profile_root/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties` file.

B. Add the following line to the file:

   ```
   uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
   ```

   Replace `portal_admin_DN` with the distinguished name of the portal administrator.

C. Save your changes and close the file.

D. Run the `./ConfigEngine.sh action-express-memberfixer -DPortalAdminPwd=password -DWasPassword=password -DModifyPortsServer=servername -DStartPort=port number` task, located in the `wp_profile_root/ConfigEngine` directory.

9. Optional: Perform the following steps if you need to change the ports for WebSphere Application Server or WebSphere Portal:

   **Note:** The starting port parameter is required for a successful completion of the `modify-ports-by-startport` task.

   Once you specify a start port, this port becomes the base for assigning port values. The code increments this value as each port is assigned, which means that the WebSphere Portal ports will be assigned incrementally starting with the port defined with the `modify-ports-by-startport` task.

A. Stop the server1 and WebSphere_Portal servers.

B. Run one of the following commands for each server you need to change:

<table>
<thead>
<tr>
<th>Method</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting port number</td>
<td><code>./ConfigEngine.sh modify-ports-by-startport -DStartPort=start_port</code></td>
</tr>
<tr>
<td></td>
<td><code>DWasPassword=password</code></td>
</tr>
<tr>
<td></td>
<td><code>DModifyPortsServer=servername</code></td>
</tr>
</tbody>
</table>

225
C. Restart the server1 and WebSphere_Portal servers.

Parent topic: Setting up a stand-alone server on AIX
Previous topic: Preparing your AIX operating system
Next topic: Configuring databases

Related concepts
Installation methods
IBM Support Assistant Lite for WebSphere Portal

Related tasks
Creating and maintaining multiple profiles on AIX
Using the member fixer tool with IBM Lotus Web Content Management

Related reference
Advanced installation parameters

---

<table>
<thead>
<tr>
<th>Port file Note: Sample port files are available on the Setup disc.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>./ConfigEngine.sh modify-ports-by-portsfile -DWasPassword=password -DModifyPortsServer=servername -DPortsFile=full path to ports file</code></td>
</tr>
<tr>
<td>The following is an example of the information within a port file although the port values will be different based on your environment:</td>
</tr>
<tr>
<td>BOOTSTRAP_ADDRESS=10031</td>
</tr>
<tr>
<td>SOAP_CONNECTOR_ADDRESS=10033</td>
</tr>
<tr>
<td>SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=10032</td>
</tr>
<tr>
<td>CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=10025</td>
</tr>
<tr>
<td>CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=10036</td>
</tr>
<tr>
<td>WC_adminhost=10027</td>
</tr>
<tr>
<td>WC_defaulthost=33344</td>
</tr>
<tr>
<td>DCS_UNICAST_ADDRESS=10029</td>
</tr>
<tr>
<td>WC_adminhost_secure=10039</td>
</tr>
<tr>
<td>WC_defaulthost_secure=10035</td>
</tr>
<tr>
<td>SIB_ENDPOINT_ADDRESS=10026</td>
</tr>
<tr>
<td>SIB_ENDPOINT_SECURE_ADDRESS=10037</td>
</tr>
<tr>
<td>SIB_MQ_ENDPOINT_ADDRESS=10030</td>
</tr>
<tr>
<td>SIB_MQ_ENDPOINT_SECURE_ADDRESS=10028</td>
</tr>
<tr>
<td>ORB_LISTENER_ADDRESS=10034</td>
</tr>
</tbody>
</table>
Configuring databases

Before transferring default data to the production database server, you need to prepare the database server. Database preparation includes creating user IDs and databases. For this installation path, stand-alone production server, remote databases servers are used.

- **Prerequisites**
  - Preparing your AIX operating system
  - Installing WebSphere Portal on AIX

Select the database server that are using.

- **Preparing DB2 on AIX for a stand-alone production server**
  You can install DB2 on the same server as WebSphere Portal, also referred to as local, or on a different server, referred to as remote. If you plan to use a local DB2 database, scripts are provided to simplify the database creation and transfer. More information about the local database is provided in the single server installation instructions. For a remote database you must complete the transfer manually.

- **Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server**
  You can install DB2 on the same server as WebSphere Portal, also referred to as local, or on a different server, referred to as remote. If you plan to use a local DB2 database, scripts are provided to simplify the database creation and transfer. More information about the local database is provided in the single server installation instructions. For a remote database you must complete the transfer manually.

- **Preparing DB2 for z/OS for a stand-alone production server**
  Setting up the DB2 for z/OS database includes creating user IDs, databases, and table spaces on a remote server.

- **Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server**
  Setting up the Oracle database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

- **Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server**
  Setting up the Oracle RAC database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

- **Setting up a remote SQL Server database on Windows for a stand-alone production server**
  To set up a remote Microsoft SQL Server Enterprise Edition database, create user IDs and databases on a remote server. Tasks are provided to assist with creating the user IDs and the databases. Before you can use the tasks, you must modify properties files.

- **Verifying databases**
  After you configure IBM WebSphere Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.
Preparing DB2 on AIX for a stand-alone production server

You can install DB2 on the same server as WebSphere Portal, also referred to as local, or on a different server, referred to as remote. If you plan to use a local DB2 database, scripts are provided to simplify the database creation and transfer. More information about the local database is provide in the single server installation instructions. For a remote database you must complete the transfer manually.

1. AIX stand-alone: Installing DB2 on AIX
   View information on installing DB2 for use with WebSphere Portal.

2. Create users
   View the steps to set up users for DB2 to work with WebSphere Portal.

3. Creating remote databases
   A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

4. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

5. Modifying database properties
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

6. Configuring JCR collation support
   View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

7. Configure WebSphere Portal to use DB2
   View information on how to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

8. Configuring DB2 for large file handling in Web Content Management
   If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the fullyMaterializeLobData property in the WebSphere Application Server administrative console.

9. Optional: Changing driver types
   WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Parent topic: Configuring databases
AIX stand-alone: Installing DB2 on AIX

View information on installing DB2 for use with WebSphere Portal.

Before you begin:

- Make sure DB2 is set up with updated kernel parameters according to the recommendations in the DB2 Quick Beginnings guide at DB2 Technical Support.

- *(For Type 2 drivers only)*: DB2 Connect must be installed on the same system as WebSphere Portal, and have the same name as the server profile name.

- When you install DB2 using the DB2 installation program, it automatically creates a DB2 administrative user with the correct operating system rights.

1. If you are using the JDBC driver in type 2 mode, configure your DB2 Connect client with the following commands. If you are using a remote database, complete this step separately from the WebSphere Portal installation.
   - `db2 update dbm cfg using tp_mon_name WAS`
   - `db2 update dbm cfg using spm_name hostname`, where `hostname` is the host name of WebSphere Portal.

   Because the default for `spm_name` is the hostname itself, specifying the `hostname` parameter is optional. If your hostname is more than eight characters, use empty double quotes (" "). For example, `db2 update dbm cfg using spm_name " "`.

2. Before installing DB2, log in with a user ID that has administrative authority.

3. To install DB2, follow the instructions that are provided with the DB2 documentation to install DB2 and the required fix pack.

Parent topic: Preparing DB2 on AIX for a stand-alone production server

Next topic: Create users
Create users

View the steps to set up users for DB2 to work with WebSphere Portal.

Before you begin: You should have completed Installing DB2.

- **Prerequisites**
  - AIX stand-alone: Installing DB2 on AIX

For all databases, use one user with administrative rights on the operating system and the DB2 installation. This user can be the database administrative user that is created automatically by the DB2 installation program. This user is the database configuration user that will be used for configuration tasks: creating database tables and performing database transfer. If you choose to have WebSphere Portal also create the databases, the database user should be given **SYSADM** rights. You only need to manually create an administrator ID when you do not want to use an existing DB2 administrator ID.

A common user name is `db2inst1`, but you can assign any user name as long as it has administrative access and follows the limitations listed here. Do not change the user name after creating it.

The limitations on user names are:

- User names can contain one to eight characters.
- Group and instance names can contain one to eight characters.
- Names cannot be any of the following: `users admin
  guests
  public
  local`
- Names cannot begin with: `IBM SQL SYS`
- Names cannot include accented characters.
- Create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Following are the basic steps to create new users and grant privileges. For more information, refer to the DB2 product documentation.

1. From the DB2 Control Center, expand the object tree until you find the DB Users folder.
2. Right-click the DB Users folder and select **Create/Add**. The Create DB User notebook opens.
3. In the User name field, type a user ID.
4. Assign an authority for the user.
5. Click **OK**.

**Parent topic:** Preparing DB2 on AIX for a stand-alone production server

**Previous topic:** AIX stand-alone: Installing DB2 on AIX

**Next topic:** Creating remote databases
Creating remote databases

A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

Before creating these databases, note the following information:

- Use IBM® DB2 Universal Database™ Enterprise Server Edition JDBC driver, JCC, type 4.
- If the DB2 JDBC type 4 driver is used, only the instructions on the remote server are required. You do not need to install the DB2 client software, and you do not need to complete the steps related to the DB2 client.
- The client software, DB2 Connect, must be correctly configured to connect to the remote DB2 server instance, for example, `db2inst1`.
- These instructions assume that a remote DB2 server and DB2 Connect are already installed and running. For information on installing DB2 Connect, see the DB2 product documentation.

**Prerequisites**

- **AIX stand-alone: Installing DB2 on AIX**
- **Create users**

To create a database, you must be a DB2 System Administrator with sufficient database privileges (SYSADM or at a minimum SYSCtrl).

1. Log in to the DB2 server system as a user with sufficient database privileges.
2. Initialize a DB2 command environment by opening a command prompt and typing `su - db2inst1`, where `db2inst1` is the database user. The prompt changes to your operating system shell prompt, for example: `$`. In this mode, you must type `db2` at the beginning of each command and use double quotation marks (""") around the entire command. The following example shows how to use the double quotes; it is not an actual command that you run:

   ```
   db2 "CREATE REGULAR TABLESPACE SMS PAGESIZE 4 K MANAGED BY SYSTEM USING ('sms') BUFFERPOOL SMSPOOL"
   ```

   **Note:** If you are using the Command Line Processor (CLP), refer to your DB2 documentation for details. The command prompt is `db2=>`. In this mode, commands can be entered without the `db2` prefix or the double quotation marks. However, the following steps assume you are not using the CLP and are entering commands from your operating system shell prompt, for example: `$`.

3. Run the following commands on the DB2 server system to configure the DB2 database instance:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DB2</strong></td>
<td><code>db2set DB2_RA_TO_RS=YES</code>&lt;br&gt;<code>db2set DB2_EVALUNCOMMITTED=YES</code>&lt;br&gt;<code>db2set DB2_INLIST_TO_NLJN=YES</code>&lt;br&gt;<code>db2 &quot;UPDATE DBM CFG USING query_heap_sz 32768&quot;</code>&lt;br&gt;<code>db2 &quot;UPDATE DBM CFG USING maxagents 500&quot;</code>&lt;br&gt;<code>db2 &quot;UPDATE DBM CFG USING sheapthres 0&quot;</code></td>
</tr>
</tbody>
</table>

4. **(Important)** Failure to complete this step will result in an unsuccessful database transfer. Run the following commands on the DB2 server system to create the necessary databases:

   - Replace `dbname` with the actual name of the database. Run the commands and each time replace `dbname` with the actual values for release, community, customization, Java Content Repository, Feedback, and Likeminds. You will need to run the commands once for each database for a total of six times.

   **Notes:**

   - Replace `dbname` with the actual name of the database. Run the commands and each time replace `dbname` with the actual values for release, community, customization, Java Content Repository, Feedback, and Likeminds. You will need to run the commands once for each database for a total of six times.

   **Remember:** DB2 database names cannot...
A. The database names should not exceed eight characters. Therefore, consider using these database names: *release, commun, custom, jcrdb, fdbkdb, and lmdb.*

```sql
db2 "CREATE DB dbname using codeset UTF-8 territory us PAGESIZE 8192"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING applheapsz 4096"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING app_ctl_heap_sz 1024"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING stmtheap 32768"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING dbheap 2400"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING locklist 1000"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING logfilsiz 4000"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING logprimary 12"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING logsecond 20"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING logbufsz 32"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING avg_appls 5"
```

```sql
db2 "UPDATE DB CFG FOR dbname using AUTO_MAINT off"
```

5. Complete the following:

A. On the DB2 server system, run the following commands. This step is only required for the IBM Java Content Repository database *(jcrdb).*

- `-jcrdb` is the name of the database used to store user data and objects
- `-jcr` is the jcr user for `jcrdb`**Note:** This value can be replaced with any ID that has administrative authority.
- `-dbpassword` is the password for the jcr user for the `jcrdb`

```sql
db2 "CONNECT TO jcrdb USER jcr USING dbpassword"
```

```sql
db2 "CREATE BUFFERPOOL ICMLSFREQBP4 SIZE 1000 PAGESIZE 4 K"
```

```sql
db2 "CREATE BUFFERPOOL ICMLSVOLATILEBP4 SIZE 16000 PAGESIZE 4 K"
```

```sql
db2 "CREATE BUFFERPOOL ICMLSMAINBP32 SIZE 16000 PAGESIZE 32 K"
```

```sql
db2 "CREATE BUFFERPOOL CMBMAIN4 SIZE 1000 PAGESIZE 4 K"
```

```sql
db2 "CREATE REGULAR TABLESPACE ICMLFQ32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLFQ32') BUFFERPOOL ICMLSMAINBP32"
```

```sql
db2 "CREATE REGULAR TABLESPACE ICMLNF32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLNF32') BUFFERPOOL ICMLSMAINBP32"
```

```sql
db2 "CREATE REGULAR TABLESPACE ICMVFPQ4 PAGESIZE 4 K MANAGED BY SYSTEM USING ('ICMVFPQ4') BUFFERPOOL ICMLSVOLATILEBP4"
```

```sql
db2 "CREATE REGULAR TABLESPACE ICMSFQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('ICMSFQ04') BUFFERPOOL ICMLSFREQBP4"
```

```sql
db2 "CREATE REGULAR TABLESPACE CMBINV04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('CMBINV04') BUFFERPOOL CMBMAIN4"
```

```sql
db2 "CREATE SYSTEM TEMPORARY TABLESPACE ICMLSYSSTSPACE32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('icmlsysstspace32') BUFFERPOOL ICMLSMAINBP32"
```

```sql
db2 "CREATE SYSTEM TEMPORARY TABLESPACE ICMLSYSSTSPACE4 PAGESIZE 4 K MANAGED BY SYSTEM USING ('icmlsysstspace4') BUFFERPOOL ICMLSVOLATILEBP4"
```

```sql
db2 "DISCONNECT jcrdb"
```

```sql
db2 "TERMINATE"
```

B. For JDBC Type 2 connections only: On DB2 Connect, use a text editor to open the `/etc/services` file. If it does not specify the DB2 connection service port, add the following text to specify the port for the remote DB2 instance:

```plaintext
db2inst1/tcp # DB2 connection service port
```

where `-db2inst1` is the name of the DB2 instance ID on the system, and `-port1` with the actual port number that is
assigned to the DB2 connection service in your DB2 server installation. The connection service port on the DB2
Client system, WebSphere Portal server, must match the connection service port on the DB2 server. The ports
should match by number but not necessarily by name.

C. Set up the correct service name by entering the following command on the DB2 server system: db2 "UPDATE DBM
CFG USING svcname svce_name" where svce_name is the connection service port name that is specified above.

D. For JDBC Type 2 connections only: On DB2 Connect, set DB2COMM to TCP/IP by using the db2set command
db2set DB2COMM=tcpip.

E. For JDBC Type 2 connections only: Catalog the TCP/IP node with the IP address of the remote database server on
DB2 Connect: db2 "catalog tcpip node remote_db_node_alias remote database_server_node server
connection_service_port" where:
- remote_db_node_alias is the alias name of the database server that you are defining for the WebSphere Application
  Server node name. The alias name can contain one to eight characters.
- database_server_node is the fully qualified host name of your database server system.
- connection_service_port is the name of the DB2 connection service port that is configured in the /etc/services
  file on the database server system.

F. For JDBC Type 2 connections only, catalog the WebSphere Portal databases on DB2 Connect, where:
- remote_db_name_domain, is the cataloged name of the databases on the server system for each domain.
- domain_alias_name, is the database alias names that you are defining.
- remote_db_node_alias is the name that was used previously when you cataloged the TCP/IP node in the previous
  step.

The alias for each database must be different from the actual database name and can only contain up to eight
characters.db2 "catalog db remote_db_name_release as release_alias_name at node remote_db_node_alias"
db2 "catalog db remote_db_name_community as comm_alias_name at node remote_db_node_alias"
db2 "catalog db remote_db_name_customization as cust_alias_name at node remote_db_node_alias"

G. For JDBC Type 2 connections only: Log out of DB2 Connect by entering db2 "terminate".

H. For JDBC Type 2 connections only, on DB2 Connect, test your remote connection by issuing the following command
in the DB2 command window: db2 "connect to alias_name user username using password", where alias_name
is the alias name that you defined above, username is the database user, and password is the password assigned to
the database user.

Parent topic: Preparing DB2 on AIX for a stand-alone production server
Previous topic: Create users
Next topic: Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root`/PortalServer/config/tablesspaces/dbdomain.space_mapping.properties file.
- The page size of table spaces used by WebSphere Portal must be 8192 bytes.
- For details on creating table spaces refer to the documentation for the database.

- **Prerequisites**
  - AIX stand-alone: Installing DB2 on AIX
  - Create users
  - Creating remote databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tablesspaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`
   
   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds

3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword `IN` and a space. For example: `community.COMP_INST.tablespace=IN COMM8KSPACE`
   
   Repeat this step for each domain that you are transferring.

4. Save and close `dbdomain.space_mapping.properties`.

5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - **Windows**: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
   - **UNIX**: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

Parent topic: Preparing DB2 on AIX for a stand-alone production server
Previous topic: Creating remote databases
Next topic: Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

**Prerequisites**
- AIX stand-alone: Installing DB2 on AIX
- Create users
- Creating remote databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:
  - release.DbName = release
  - jcr.DbName = jcrdb
  - feedback.DbName = fdbkdb
  - likeminds.DbName = lmdb
  - community.DbName = commdb
  - customization.DbName = custdb
- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
- **dbdomainDbType**
- **dbdomainDbName**
- **dbdomainDbUrl**
- **dbdomainDbSchema**

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If **DbUser**, **DbUrl**, and **DbPassword** are not the same across domains, the value for **DataSourceName** must differ from the **DataSourceName** of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   
   - **wp_profile_root/ConfigEngine/properties/wkplc.properties**
   - **wp_profile_root/ConfigEngine/properties/wkplc_comp.properties**
   - **wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties**

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In **wkplc_comp.properties**, most properties are repeated for each domain.

2. Use a text editor to open the properties file **wkplc_comp.properties** and modify the values to correspond to your environment.
   
   A. For **dbdomainDbType**, type *db2*.
   
   B. For **dbdomainDbName**, type the name of the WebSphere Portal domain database. **Notes:**
      
      - This value is also the database element in the **dbdomainDbUrl** property.
      
      - This value is the TCP-IP alias for the database.

   C. For **dbdomainDbSchema**, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.

   D. For **dbdomainDataSourceName**, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback

   E. For **dbdomainDbUrl**, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:** The database element of this value should match the value of **DbName**.

   F. For **dbdomainDbUser**, type the user ID for the database administrator.

   G. For **dbdomainDbPassword**, type the password for the database administrator.

   H. Optional: For **dbdomainDbRuntimeUser**, type the user ID of the database user that should be used by WebSphere Portal to connect to the database at runtime.

   I. If **dbdomainDbRuntimeUser** is specified, you must set **dbdomainDbRuntimePassword** to be the password of the runtime database user.

   J. For **dbdomainDBA.DbUser**, type the database administrator user ID for privileged access operations during creation of the database.
K. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.

L. For `dbdomain.XDbName`, type the database loop back alias that needs to be set if you plan to use the `create-database` task. **Note:** Required only for local databases using a JDBC Type 2 driver.

M. For `dbdomain.DbNode`, type the value for the node database. Set this value if you want to call `create-database`. **Note:** Required only for local databases.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For `source.domain.DbType`, type of the database you are currently configured to use. The value for `source.domain(DbType)` is Derby by default.

B. For `source.domain.DbName`, type the name of the database domain you are currently using.

C. For `source.domain.DbSchema`, type current schema identifier for objects within the database for this domain.

D. For `source.domain.DataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For `source.domain.DbUrl`, type the url currently used to access your database.

F. For `source.domain.DbUser`, type the name of the user accessing this database.

G. For `source.domain.DbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

   A. For `db2.DbDriver`, type the name of the JDBC driver class.

   B. For `db2.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For `db2.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

---

**Parent topic:** Preparing DB2 on AIX for a stand-alone production server  
**Previous topic:** Assigning custom table spaces  
**Next topic:** Configuring JCR collation support
Configuring JCR collation support

View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

- **Prerequisites**
  - AIX stand-alone: Installing DB2 on AIX
  - Create users
  - Creating remote databases
  - Modifying database properties

1. Stop the WebSphere Portal server:

   ```
   . ./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password
   ```

2. If using a remote DB2 server, copy the following files from the WebSphere Portal server to a temporary directory on the remote DB2 server:

   ```
   PortalServer_root/jcr/prereq.jcr/config/collation.jar
   PortalServer_root/jcr/prereq.jcr/config/registerCollationUDFTemplate.sql
   ```

3. Set up collation on the database where the JCR domain is located.

   A. Change to the directory `db2_instance_owner_home/sqllib/function`.

   B. Execute the command: Remote DB2:

   ```
   db2_instance_owner_home/sqllib/java/jdk/bin/jar -xvf temporary_location/collation.jar icm/CollationUDF.class
   ```

   Local DB2:

   ```
   db2_instance_owner_home/sqllib/java/jdk/bin/jar -xvf PortalServer_root/jcr/prereq.jcr/config/collation.jar icm/CollationUDF.class
   ```

   C. If using a local DB2, change to the directory `wp_profile_root/PortalServer/jcr/config`. If using a remote DB2, change to the temporary directory where you copied the files in a previous step.

   D. Open the file `registerCollationUDFTemplate.sql` and change all `SCHEMA` references to the JCR schema; for example, `JCR`. The value set for `SCHEMA` should match the value set for the `jcr.DbSchema` property. You specify `jcr.DbSchema` in the configuration file `wkplc_comp.properties` when you modify database properties.

   E. Connect to the JCR database by running:

   ```
   db2 connect to <jcrdb> user <userid> using <password>
   ```

   F. Execute the script by running the command: Remote DB2:

   ```
   db2 -tvf temporary_location/registerCollationUDFTemplate.sql
   ```

   Local DB2:

   ```
   db2 -tvf wp_profile_root/PortalServer/jcr/config/registerCollationUDFTemplate.sql
   ```

4. Verify that the UDF is registered properly.

   A. Log in as the `db2instanceID`. Open a DB2 terminal window, and type `db2`. From the command line, type the command:

   ```
   connect to JCRDB user userid using password
   ```

   You must connect to the JCR database using the userid and password of the WebSphere Portal JCR data source.

   B. Execute the following command in the previous substep. If the command completes successfully, the UDF is registered correctly as follows:

   ```
   values schema.sortkeyj(‘abc’,’en’), where schema is the schema used in the previous substep.
   ```

5. Edit the `icm.properties` file, located in `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm` directory. Add the following section to the end of the file: Enable/Disable collation support for all DB2 platforms.
# Disabled by default
jcr.query.collation.db2.enabled = true

# Database specific collation mappings
# These mappings apply map a Java locale name into a collation name
# supported by the underlying database.
# Example mappings for DB2 platform

# English
jcr.query.collation.en = en

# Swedish
jcr.query.collation.sv = sv

jcr.query.collation.zh = zh
jcr.query.collation.de = de
jcr.query.collation.da = da
jcr.query.collation.hu = hu
jcr.query.collation.jp = jp

6. Enter the following command to start the WebSphere Portal server: `./startServer.sh WebSphere_Portal`
Configure WebSphere Portal to use DB2

View information on how to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

- Prerequisites
  - AIX stand-alone: Installing DB2 on AIX
  - Create users
  - Creating remote databases
  - Modifying database properties

Tips:
- To run these tasks as a non-root user, you must first run the task chown -R non-root_user WebSphereDir.
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

Steps for transferring data to another supported database

1. If you are running a type 2 connection, before transferring data edit the db2cli.ini file. Failure to follow these steps will cause the database transfer to hang at the task action-process-constraints.
   A. Locate the file /home/db2inst1/sqllib/cfg/db2cli.ini.
   B. Add the following to the end of the file. Leave an empty line after ReturnAliases=0.[COMMON]

   
   [COMMON]
   DYNAMIC=1
   ReturnAliases=0

2. Perform this step only if you are installing multiple instances of WebSphere Portal. Change the maximum number of databases MAX_NETBIOS_CONNECTIONS to increase the default configured number of databases. For example, enter the following command at the database prompt: set client MAX_NETBIOS_CONNECTIONS 254 A message indicates success if the number was increased.

3. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.

4. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2.

   /home/db2inst1/sqllib/db2profile

   where db2inst1 represents your database instance

   Note: You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.

5. Enter the following commands to validate configuration properties.

   . /ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
6. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.

7. Stop both WebSphere Application Server and the WebSphere Portal server:

```
Option                Description
WebSphere Application Server
    ./stopServer.sh server1 -username admin_userid -password admin_password
WebSphere Portal
    ./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password
```

8. Transfer the database:

   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory `wp_profile_root/ConfigEngine`.

   B. Enter the following command: `./ConfigEngine.sh database-transfer -DWasPassword=password`

   **Note:**
   - To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command: `./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password`
   - If you have been storing data in IBM® Derby for a long time, database transfer could fail with OutOfMemory exceptions. If database transfer fails, add the following property to the command in this step: `./ConfigEngine.sh database-transfer -DBbtJavaMaxMemory=1536M -DWasPassword=password`

   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

9. After transferring the database tables, perform a reorg check to improve performance. Perform this step for each database alias in the property file.

   A. Connect to a database with the following command:

      ```
      db2 connect to database_alias user db2admin_userid using password
      ```

      **Note:** Additional options might be required if additional security has been installed. Refer to DB2 Universal Database commands by example for links to the command reference.

   B. After it is connected, run the following commands from the DB2 prompt:

      ```
      db2 reorgchk update statistics on table all > xyz.out
      ```

      C. Look in the reorg column for entries marked with a * (star or asterisk) in the file `xyz.out`. For each line with a *, note the `tablename` and run the following command for each `tablename`:

      ```
      db2 reorg table tablename db2 terminate db2rbind database_name -l db2rbind.out -u db2_admin -p password
      ```

      D. The output file `db2rbind.out` is only created when there is an error for the `db2rbind` command.

10. Change to the directory `wp_profile_root/bin`.

11. Enter the following command to start the WebSphere Portal server:

    ```
    ./startServer.sh WebSphere_Portal
    ```

---

**Parent topic:** Preparing DB2 on AIX for a stand-alone production server  
**Previous topic:** Configuring JCR collation support  
**Next topic:** Configuring DB2 for large file handling in Web Content Management
Configuring DB2 for large file handling in Web Content Management

If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

**Prerequisites**

- AIX stand-alone: Installing DB2 on AIX
- Create users
- Creating remote databases
- Modifying database properties
- Configure WebSphere Portal to use DB2

**Note:** You only need to perform these steps if you are using Web Content Management.

1. Log into the WebSphere Application Server administrative console.
2. Click **Resources > JDBC > Data sources.**
3. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
4. Select the name of the data source that is defined in `wkplc_comp.properties` for the JCR database domain. The default data source is `wpdbDS`.
5. Click **Custom properties.**
6. Ensure that the `fullyMaterializeLobData` property is set to `false`.

**Parent topic:** Preparing DB2 on AIX for a stand-alone production server

**Previous topic:** Configure WebSphere Portal to use DB2

**Next topic:** Changing driver types
Changing driver types

WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Before you begin, ensure that the following conditions are met:

- The WebSphere Portal database has been successfully transferred to DB2 using the database-transfer configuration task.
- The files wkplc_comp.properties and wkplc_dbtype.properties have been modified to set the correct values for the DB2 drivers that you are switching to: In the file wkplc_comp.properties set each <Domain>.DbUrl property using the following formats:

  # db2 (type 2):        { jdbc:db2:wpsdb }
  # db2 (type 4):        { jdbc:db2://<YourDatabaseServer>:50000/wpsdb:returnAlias=0; }

- In the file wkplc_dbtype.properties set the db2.DbLibrary property using the following format:

  # For DB2 Type 2 driver use <SQLLIB>/java/db2java.zip
  # For DB2 Type 4 driver use <SQLLIB>/java/db2jcc.jar:<SQLLIB>/java/db2jcc_license_cu.jar

- In the file wkplc_dbtype.properties set the db2.DbDriver property using the following format:

  # For DB2 Type 2 driver use COM.ibm.db2.jdbc.app.DB2Driver
  # For DB2 Type 4 driver use com.ibm.db2.jcc.DB2Driver

**Note:** If WebSphere Portal is installed on the same machine as the DB2 server and you switch from a JDBC Type 4 connection to a JDBC Type 2 connection, verify that you have created the alias names for the DB2 databases as described in Creating remote databases and that the alias names are specified for the databases in the file wkplc_comp.properties. When switching from a JDBC Type 2 connection to a JDBC Type 4 connection, remove the database alias names and refer to the databases directly. This is required because of a limitation in the DB2 Universal JDBC driver.

**Prerequisites**

- AIX stand-alone: Installing DB2 on AIX
- Create users
- Creating remote databases
- Modifying database properties
- Configure WebSphere Portal to use DB2

1. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.
2. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2.

   ```bash
   /home/db2inst1/sqllib/db2profile
   ```

   where db2inst1 represents your database instance

   **Note:** You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.

3. Enter the following commands to validate configuration properties.

   ```bash
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,
   ```
4. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td><code>./stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td><code>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

6. Change to the directory `wp_profile_root/ConfigEngine`.

7. To change from one supported driver to the other, run the following task to connect the database, including only the domains that require the switch:

   ```bash
   ```

8. Change to the directory `wp_profile_root/bin`.

9. Enter the following command to start the WebSphere Portal server:

   ```bash
   ./startServer.sh WebSphere_Portal
   ```

**Parent topic:** Preparing DB2 on AIX for a stand-alone production server

**Previous topic:** Configuring DB2 for large file handling in Web Content Management
Preparing DB2 on HP-UX, Linux, and Solaris for a standalone production server

You can install DB2 on the same server as WebSphere Portal, also referred to as local, or on a different server, referred to as remote. If you plan to use a local DB2 database, scripts are provided to simplify the database creation and transfer. More information about the local database is provided in the single server installation instructions. For a remote database you must complete the transfer manually.

1. **Installing DB2**
   View information on installing DB2 for use with WebSphere Portal.

2. **Create users**
   View the steps to set up users for DB2 to work with WebSphere Portal.

3. **Creating remote databases**
   A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

4. **Optional: Assigning custom table spaces**
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

5. **Modifying database properties**
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

6. **Configuring JCR collation support**
   View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

7. **Configure WebSphere Portal to use DB2**
   View the steps to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

8. **Configuring DB2 for large file handling in Web Content Management**
   If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

9. **Optional: Changing driver types**
   WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

**Parent topic:** Configuring databases
Installing DB2

View information on installing DB2 for use with WebSphere Portal.

Before you begin:

- Make sure DB2 is set up with updated kernel parameters according to the recommendations in the DB2 Quick Beginnings guide at DB2 Technical Support.
- **(For Type 2 drivers only):** DB2 Connect must be installed on the same system as WebSphere Portal, and have the same name as the server profile name.
- When you install DB2 using the DB2 installation program, it automatically creates a DB2 administrative user with the correct operating system rights.

1. If you are using the JDBC driver in type 2 mode, configure your DB2 Connect client with the following commands. If you are using a remote database, complete this step separately from the WebSphere Portal installation.
   - `db2 update dbm cfg using tp_mon_name WAS`
   - `db2 update dbm cfg using spm_name hostname`, where `hostname` is the host name of WebSphere Portal.
   Because the default for `spm_name` is the hostname itself, specifying the `hostname` parameter is optional. If your hostname is more than eight characters, use empty double quotes (" "). For example, `db2 update dbm cfg using spm_name " "`.

2. Before installing DB2, log in with a user ID that has administrative authority. The system user ID and password must match the database user ID and password for the WebSphere Portal databases. If you do not use the default DB2 database user ID, or if you must access a remote database, create the system user ID before installing DB2.
   To set up the environment for the database access:
   A. The initialization script for this user (for example, `user-home/.profile`) must contain a call to the `db2profile` script in the `db-home/sql/lib` directory.
   B. After you create the system user ID and password for the DB2 installation, add the user ID to the DB2 administration group (such as `db2adm`) for that system.

   To ensure that the user is set up correctly, login with this user ID and start the DB2 command line processor by executing the command: `db2`. If the command line processor displays, the user ID is set up correctly.

3. To install DB2, follow the instructions that are provided with the DB2 documentation to install DB2 and the required fix pack.

Parent topic: Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server
Next topic: Create users
Create users

View the steps to set up users for DB2 to work with WebSphere Portal.

Before you begin: You should have completed Installing DB2.

- Prerequisites
  - Installing DB2

For all databases, use one user with administrative rights on the operating system and the DB2 installation. This user can be the database administrative user that is created automatically by the DB2 installation program. This user is the database configuration user that will be used for configuration tasks: creating database tables and performing database transfer. If you choose to have WebSphere Portal also create the databases, the database user should be given SYSADM rights. You only need to manually create an administrator ID when you do not want to use an existing DB2 administrator ID.

A common user name is db2inst1, but you can assign any user name as long as it has administrative access and follows the limitations listed here. Do not change the user name after creating it.

The limitations on user names are:
- User names can contain one to eight characters.
- Group and instance names can contain one to eight characters.
- Names cannot be any of the following: users, admins, guests, public, local.
- Names cannot begin with: IBM, SQL, SYS.
- Names cannot include accented characters.
- Create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Following are the basic steps to create new users and grant privileges. For more information, refer to the DB2 product documentation.

1. From the DB2 Control Center, expand the object tree until you find the DB Users folder.
2. Right-click the DB Users folder and select Create/Add. The Create DB User notebook opens.
3. In the User name field, type a user ID.
4. Assign an authority for the user.
5. Click OK.

Parent topic: Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server
Previous topic: Installing DB2
Next topic: Creating remote databases
Creating remote databases

A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

Before creating these databases, note the following information:

- Use IBM® DB2 Universal Database™ Enterprise Server Edition JDBC driver, JCC, type 4.
- If the DB2 JDBC type 4 driver is used, only the instructions on the remote server are required. You do not need to install the DB2 client software, and you do not need to complete the steps related to the DB2 client.
- The client software, DB2 Connect, must be correctly configured to connect to the remote DB2 server instance, for example, db2inst1.
- These instructions assume that a remote DB2 server and DB2 Connect are already installed and running. For information on installing DB2 Connect, see the DB2 product documentation.

**Prerequisites**

- Installing DB2
- Create users

To create a database, you must be a DB2 System Administrator with sufficient database privileges (SYSADM or at a minimum SYSTCTRL).

1. Log in to the DB2 server system as a user with sufficient database privileges.

2. Initialize a DB2 command environment by opening a command prompt and typing `su - db2inst1`, where `db2inst1` is the database user. The prompt changes to your operating system shell prompt, for example: `$`. In this mode, you must type `db2` at the beginning of each command and use double quotation marks (`"`) around the entire command. The following example shows how to use the double quotes; it is not an actual command that you run:

   `db2 "CREATE REGULAR TABLESPACE SMS PAGESIZE 4 K MANAGED BY SYSTEM USING ('sms') BUFFERPOOL SMSPOOL"

   **Note:** If you are using the Command Line Processor (CLP), refer to your DB2 documentation for details. The command prompt is `db2=>`. In this mode, commands can be entered without the `db2` prefix or the double quotation marks. However, the following steps assume you are not using the CLP and are entering commands from your operating system shell prompt, for example: `$`.

3. Run the following commands on the DB2 server system to configure the DB2 database instance:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td><code>db2set DB2_RA_TO_RS=YES</code>&lt;br&gt;<code>db2set DB2_EVALUNCOMMITTED=YES</code>&lt;br&gt;<code>db2set DB2_INLIST_TO_NLJN=YES</code>&lt;br&gt;<code>db2 &quot;UPDATE DBM CFG USING query_heap_sz 32768&quot;</code>&lt;br&gt;<code>db2 &quot;UPDATE DBM CFG USING maxagents 500&quot;</code>&lt;br&gt;<code>db2 &quot;UPDATE DBM CFG USING sheapthres 0&quot;</code></td>
</tr>
</tbody>
</table>

4. **(Important)** Failure to complete this step will result in an unsuccessful database transfer. Run the following commands on the DB2 server system to create the necessary databases:

   **Notes:**
   - Replace `dbname` with the actual name of the database. Run the commands and each time replace `dbname` with the actual values for release, community, customization, Java Content Repository, Feedback, and Likeminds. You will need to run the commands once for each database for a total of six times.
   **Remember:** DB2 database names cannot
exceed eight characters. Therefore, consider using these database names: release, commun, custom, jcrdb, fdbkdb, and lmdb.

```sql
db2 "CREATE DB dbname using codeset UTF-8 territory us PAGESIZE 8192"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING applheapsz 4096"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING app_ctl_heap_sz 1024"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING stmtheap 32768"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING dbheap 2400"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING locklist 1000"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING logfilsiz 4000"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING logprimary 12"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING logsecond 20"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING logbufsz 32"
```

```sql
db2 "UPDATE DB CFG FOR dbname USING avg_appls 5"
```

```sql
db2 "UPDATE DB CFG FOR dbname using AUTO_MAINT off"
```

5. Complete the following:

A. On the DB2 server system, run the following commands. This step is only required for the IBM Java Content Repository database (jcrdb).

- jcrdb is the name of the database used to store user data and objects
- jcr is the jcr user for jcrdb

Note: This value can be replaced with any ID that has administrative authority.

- dbpassword is the password for the jcr user for the jcrdb

```sql
db2 "CONNECT TO jcrdb USER jcr USING dbpassword"
```

```sql
db2 "CREATE BUFFERPOOL ICMLSFREQBP4 SIZE 1000 PAGESIZE 4 K"
```

```sql
db2 "CREATE BUFFERPOOL ICMLSVOLATILEBP4 SIZE 16000 PAGESIZE 4 K"
```

```sql
db2 "CREATE BUFFERPOOL ICMLSMAINBP32 SIZE 16000 PAGESIZE 32 K"
```

```sql
db2 "CREATE BUFFERPOOL CMBMAIN4 SIZE 1000 PAGESIZE 4 K"
```

```sql
db2 "CREATE REGULAR TABLESPACE ICMLFQ32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLFQ32') BUFFERPOOL ICMLSMAINBP32"
```

```sql
db2 "CREATE REGULAR TABLESPACE ICMLNF32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLNF32') BUFFERPOOL ICMLSMAINBP32"
```

```sql
db2 "CREATE REGULAR TABLESPACE ICMVFPQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('ICMVFPQ04') BUFFERPOOL ICMLSVOLATILEBP4"
```

```sql
db2 "CREATE REGULAR TABLESPACE ICMSFPQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('ICMSFPQ04') BUFFERPOOL ICMLSVOLATILEBP4"
```

```sql
db2 "CREATE REGULAR TABLESPACE CMBINV04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('CMBINV04') BUFFERPOOL CMBMAIN4"
```

```sql
db2 "CREATE SYSTEM TEMPORARY TABLESPACE ICMLSYSSTSPACE32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('icmlsysstspace32') BUFFERPOOL ICMLSMAINBP32"
```

```sql
db2 "CREATE SYSTEM TEMPORARY TABLESPACE ICMLSYSSTSPACE4 PAGESIZE 4 K MANAGED BY SYSTEM USING ('icmlsysstspace4') BUFFERPOOL ICMLSVOLATILEBP4"
```

```sql
db2 "DISCONNECT jcrdb"
```

```sql
db2 "TERMINATE"
```

B. For JDBC Type 2 connections only: On DB2 Connect, use a text editor to open the /etc/services file. If it does not specify the DB2 connection service port, add the following text to specify the port for the remote DB2 instance:

```text
$db2inst1/tcp # DB2 connection service port
```

where $db2inst1$ is the name of the DB2 instance ID on the system, and $port$ with the actual port number that is
assigned to the DB2 connection service in your DB2 server installation. The connection service port on the DB2
Client system, WebSphere Portal server, must match the connection service port on the DB2 server. The ports
should match by number but not necessarily by name.

C. Set up the correct service name by entering the following command on the DB2 server system:
   `db2 "UPDATE DBM CFG USING svcename svce_name"` where `svce_name` is the connection service port name that is specified above.

D. For JDBC Type 2 connections only: On DB2 Connect, set DB2COMM to TCP/IP by using the db2set command
   `db2set DB2COMM=tcpip`.

E. For JDBC Type 2 connections only: Catalog the TCP/IP node with the IP address of the remote database server on
   DB2 Connect:
   `db2 "catalog tcpip node remote_db_node_alias remote database_server_node server connection_service_port"` where:
   - `remote_db_node_alias` is the alias name of the database server that you are defining for the WebSphere Application
     Server node name. The alias name can contain one to eight characters.
   - `database_server_node` is the fully qualified host name of your database server system.
   - `connection_service_port` is the name of the DB2 connection service port that is configured in the `/etc/services`
     file on the database server system.

F. For JDBC Type 2 connections only, catalog the WebSphere Portal databases on DB2 Connect, where:
   - `remote_db_name_domain`, is the cataloged name of the databases on the server system for each domain.
   - `domain_alias_name`, is the database alias names that you are defining.
   - `remote_db_node_alias` is the name that was used previously when you cataloged the TCP/IP node in the previous
     step.
   
   The alias for each database must be different from the actual database name and can only contain up to eight
   characters:
   - `db2 "catalog db remote_db_name_release as release_alias_name at node remote_db_node_alias"
   - `db2 "catalog db remote_db_name_community as comm_alias_name at node remote_db_node_alias"
   - `db2 "catalog db remote_db_name_customization as cust_alias_name at node remote_db_node_alias"
   - `db2 "catalog db remote_db_name_fdbkdb as fdbkdb_alias_name at node remote_db_node_alias"
   - `db2 "catalog db remote_db_name_lmdb as lmdb_alias_name at node remote_db_node_alias"
   - `db2 "catalog db remote_db_name_jcrdb as jcrdb_alias_name at node remote_db_node_alias"

G. For JDBC Type 2 connections only: Log out of DB2 Connect by entering `db2 "terminate"`.

H. For JDBC Type 2 connections only, on DB2 Connect, test your remote connection by issuing the following command
   in the DB2 command window: `db2 "connect to alias_name user username using password"`, where `alias_name`
   is the alias name that you defined above, `username` is the database user, and `password` is the password assigned to
   the database user.

Parent topic: Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Create users

Next topic: Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties` file.
- The page size of table spaces used by WebSphere Portal must be 8192 bytes.
- For details on creating table spaces refer to the documentation for the database.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties` that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`

   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds

3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword `IN` and a space. For example: `community.COMP_INST.tablespace=IN COMM8KSPACE`

   Repeat this step for each domain that you are transferring.

4. Save and close `dbdomain.space_mapping.properties`.

5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - **Windows:** `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
   - **UNIX:** `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

**Parent topic:** Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server  
**Previous topic:** Creating remote databases  
**Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

Tip: For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Working with properties files:
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:
  - release_DbName=release
  - jcr_DbName=jcrdb
  - feedback_DbName=fdbkdb
  - likeminds_DbName=lmdb
  - community_DbName=commdb
  - customization_DbName=custdb
- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.
   A. For dbdomainDbType, type db2.
   B. For dbdomainDbName, type the name of the WebSphere Portal domain database.
      Notes:
      - This value is also the database element in the dbdomainDbUrl property.
      - This value is the TCP-IP alias for the database.
   C. For dbdomainDbSchema, type the schema name of the database domain.
      Note: Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   D. For dbdomainDataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For dbdomainDbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database.
      Note: The database element of this value should match the value of DbName.
   F. For dbdomainDbUser, type the user ID for the database administrator.
   G. For dbdomainDbPassword, type the password for the database administrator.
   H. Optional: For dbdomainDbRuntimeUser, type the user ID of the database user that should be used by WebSphere Portal to connect to the database at runtime.
   I. If dbdomainDbRuntimeUser is specified, you must set dbdomainDbRuntimePassword to be the password of the runtime database user.
   J. For dbdomainDBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.
K. For dbdomain.DBA.DbPassword, type the database administrator password for privileged access operations during creation of the database.

L. For dbdomain.XDbName, type the database loop back alias that needs to be set if you plan to use the create-database task. **Note:** Required only for local databases using a JDBC Type 2 driver.

M. For dbdomain.DbNode, type the value for the node database. Set this value if you want to call create-database. **Note:** Required only for local databases.

3. **(Optional) When transferring WebSphere Portal databases from any database other than the default Derby database,** update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For source.domainDbType, type of the database you are currently configured to use. The value for source.domainDbType is Derby by default.

B. For source.domainDbName, type the name of the database domain you are currently using.

C. For source.domainDbSchema, type current schema identifier for objects within the database for this domain.

D. For source.domainDataSourceName, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For source.domainDbUrl, type the url currently used to access your database.

F. For source.domainDbUser, type the name of the user accessing this database.

G. For source.domainDbPassword, type the password of the source DbUser.

4. Save and close the file.

5. **Update the following properties in the file wkplc_dbtype.properties.**

   A. For db2.DbDriver, type the name of the JDBC driver class.

   B. For db2.DbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For db2.JdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. **Update the following property in the file wkplc.properties.**

   A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server  
**Previous topic:** Assigning custom table spaces  
**Next topic:** Configuring JCR collation support
Configuring JCR collation support

View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties

1. Stop the WebSphere Portal server:
   ```
   ./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password
   ```

2. If using a remote DB2 server, copy the following files from the WebSphere Portal server to a temporary directory on the remote DB2 server:
   ```
   PortalServer_root/jcr/prereq.jcr/config/collation.jar
   PortalServer_root/jcr/prereq.jcr/config/registerCollationUDFTemplate.sql
   ```

3. Set up collation on the database where the JCR domain is located.
   A. Change to the directory `db2_instance_owner_home/sqllib/function`.
   B. Execute the command: **Remote DB2**:
      ```
      db2_instance_owner_home/sqllib/java/jdk/bin/jar -xvf temporary location/collation.jar icm/CollationUDF.class
      ```
      **Local DB2**:
      ```
      db2_instance_owner_home/sqllib/java/jdk/bin/jar -xvf PortalServer_root/jcr/prereq.jcr/config/collation.jar icm/CollationUDF.class
      ```
   C. If using a local DB2, change to the directory `wp_profile_root/PortalServer/jcr/config`. If using a remote DB2, change to the temporary directory where you copied the files in a previous step.
   D. Open the file `registerCollationUDFTemplate.sql` and change all `SCHEMA` references to the JCR schema; for example, `JCR`. The value set for `SCHEMA` should match the value set for the `jcr.DbSchema` property. You specify `jcr.DbSchema` in the configuration file `wkplc_comp.properties` when you modify database properties.
   E. Connect to the JCR database by running `db2 connect to <jcrdb> user <userid> using <password>`. You must connect to the JCR database using the userid and password of the WebSphere Portal JCR data source.
   F. Execute the following command. **Remote DB2**:
      ```
      db2 -tvf temporary location/registerCollationUDFTemplate.sql
      ```
      **Local DB2**:
      ```
      db2 -tvf wp_profile_root/PortalServer/jcr/config/registerCollationUDFTemplate.sql
      ```
   G. Disconnect from the JCR database.
   H. Restart the DB2 instance.

4. Verify that the UDF is registered properly.
   A. Log in as the `db2instanceID`. Open a DB2 terminal window, and type `db2`. From the command line, type the command `connect to <jcrdb> user <userid> using <password>`. You must connect to the JCR database using the userid and password of the WebSphere Portal JCR data source.
   B. Execute the following command in the previous substep. If the command completes successfully, the UDF is registered correctly as follows: `values schema.sortkeyj('abc','en')`, where `schema` is the schema used in the previous substep.

5. Edit the `icm.properties` file, located in `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm` directory. Add the following section to the end of the file:
   ```
   # Enable/Disable collation support for all DB2 platforms
   ```
# Disabled by default

jcr.query.collation.db2.enabled = true

# Database specific collation mappings

# These mappings apply map a Java locale name into a collation name
# supported by the underlying database.
# Example mappings for DB2 platform

# English
jcr.query.collation.en = en

# Swedish
jcr.query.collation_sv = sv

jcr.query.collation.zh = zh
jcr.query.collation.de = de
jcr.query.collation.da = da
jcr.query.collation.hu = hu
jcr.query.collation.jp = jp

6. Enter the following command to start the WebSphere Portal server:

```bash
./startServer.sh WebSphere_Portal
```

Parent topic: Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Modifying database properties

Next topic: Configure WebSphere Portal to use DB2
Configure WebSphere Portal to use DB2

View the steps to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

**Prerequisites**
- Installing DB2
- Create users
- Creating remote databases
- Modifying database properties

**Tips:**
- To run these tasks as a non-root user, you must first run the task chown -R non-root_user WebSphereDir.
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

**Steps for transferring data to another supported database**

1. If you are running a type 2 connection, before transferring data edit the db2cli.ini file. Failure to follow these steps will cause the database transfer to hang at the task action-process-constraints.
   A. Locate the file /home/db2inst1/sqllib/cfg/db2cli.ini.
   B. Add the following to the end of the file. Leave an empty line after ReturnAliases=0.[COMMON]
      ```
      DYNAMIC=1
      ReturnAliases=0
      ```

2. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.

3. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2. .
   ```
   /home/db2inst1/sqllib/db2profile
   where db2inst1 represents your database instance
   ```
   **Note:** You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.

4. Enter the following commands to validate configuration properties...
   ```
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```

5. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.
6. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td>./stopServer.sh server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

7. Transfer the database:

**Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

A. Change to the directory `wp_profile_root/ConfigEngine`.

B. Enter the following command:

```
./ConfigEngine.sh database-transfer -DWasPassword=password
```

**Note:**
- To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:

```
./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
```
- If you have been storing data in IBM® Derby for a long time, database transfer could fail with OutOfMemory exceptions. If database transfer fails, add the following property to the command in this step:

```
./ConfigEngine.sh database-transfer -DDbtJavaMaxMemory=1536M -DWasPassword=password
```

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

8. After transferring the database tables, perform a `reorg` check to improve performance. Perform this step for each database alias in the property file.

A. Connect to a database with the following command:

```
db2 connect to database_alias user db2admin_userid using password
```

**Note:** Additional options might be required if additional security has been installed. Refer to DB2 Universal Database commands by example for links to the command reference.

B. After it is connected, run the following commands from the DB2 prompt:

```
db2 reorgchk update statistics on table all > xyz.out
```

C. Look in the `reorg` column for entries marked with a * (star or asterisk) in the file `xyz.out`. For each line with a *, note the `tablename` and run the following command for each `tablename`:

```
db2 reorg table tablename db2 terminate db2rbind database_name -l db2rbind.out -u db2_admin -p password
```

D. The output file `db2rbind.out` is only created when there is an error for the `db2rbind` command.

9. Change to the directory `wp_profile_root/bin`.

10. Enter the following command to start the WebSphere Portal server:

```
./startServer.sh WebSphere_Portal
```

**Parent topic:** Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Configuring JCR collation support

**Next topic:** Configuring DB2 for large file handling in Web Content Management
Configuring DB2 for large file handling in Web Content Management

If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties
  - Configure WebSphere Portal to use DB2

  **Note:** You only need to perform these steps if you are using Web Content Management.

1. Log into the WebSphere Application Server administrative console.
2. Click **Resources > JDBC > Data sources**.
3. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
4. Select the name of the data source that is defined in `wkplc_comp.properties` for the JCR database domain. The default data source is `wpdbDS`.
5. Click **Custom properties**.
6. Ensure that the `fullyMaterializeLobData` property is set to `false`.

**Parent topic:** Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Configure WebSphere Portal to use DB2

**Next topic:** Changing driver types
Changing driver types

WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Before you begin, ensure that the following conditions are met:

- The WebSphere Portal database has been successfully transferred to DB2 using the `database-transfer` configuration task.
- The files `wkplc_comp.properties` and `wkplc_dbtype.properties` have been modified to set the correct values for the DB2 drivers that you are switching to:

  In the file `wkplc_comp.properties` set each `<Domain>.DbUrl` property using the following formats:

  ```
  # db2 (type 2):        { jdbc:db2:wpsdb }
  # db2 (type 4):        { jdbc:db2://<YourDatabaseServer>:50000/wpsdb:returnAlias=0; }
  ```

  In the file `wkplc_dbtype.properties` set the `db2.DbLibrary` property using the following format:

  ```
  # For DB2 Type 2
driver use <SQLLIB>/java/db2java.zip
  # For DB2 Type 4 driver use <SQLLIB>/java/db2jcc.jar:<SQLLIB>/java/db2jcc_license_cu.jar
  ```

  In the file `wkplc_dbtype.properties` set the `db2.DbDriver` property using the following format:

  ```
  # For DB2 Type 2
driver use COM.ibm.db2.jdbc.app.DB2Driver
  # For DB2 Type 4 driver use com.ibm.db2.jcc.DB2Driver
  ```

**Note:** If WebSphere Portal is installed on the same machine as the DB2 server and you switch from a JDBC Type 4 connection to a JDBC Type 2 connection, verify that you have created the alias names for the DB2 databases as described in Creating remote databases and that the alias names are specified for the databases in the file `wkplc_comp.properties`. When switching from a JDBC Type 2 connection to a JDBC Type 4 connection, remove the database alias names and refer to the databases directly. This is required because of a limitation in the DB2 Universal JDBC driver.

**Prerequisites**

- Installing DB2
- Create users
- Creating remote databases
- Modifying database properties
- Configure WebSphere Portal to use DB2

1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`
2. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2.

   ```
   /home/db2inst1/sql/lib/db2profile
   ```

   where `db2inst1` represents your database instance. **Note:** You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.

3. Enter the following commands to validate configuration properties:

   ```
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,
   ```
4. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td><code>./stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td><code>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

6. Change to the directory `wp_profile_root/ConfigEngine`.

7. To change from one supported driver to the other, run the following task to connect the database, including only the domains that require the switch:

   ```bash
   ``

8. Change to the directory `wp_profile_root/bin`.

9. Enter the following command to start the WebSphere Portal server:

   ```bash
   ./startServer.sh WebSphere_Portal
   ``

---

**Parent topic:** Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Configuring DB2 for large file handling in Web Content Management
Preparing DB2 for z/OS for a stand-alone production server

Setting up the DB2 for z/OS database includes creating user IDs, databases, and table spaces on a remote server.

1. Installing DB2 for z/OS
   View information on how to install DB2 for z/OS for use with IBM® WebSphere® Portal.

2. Creating users
   Complete this procedure to set up users for IBM DB2 Universal Database™ for z/OS® to work with IBM WebSphere Portal.

3. Creating remote databases
   A remote database resides on a different machine than WebSphere Portal. You must manually create the required databases before configuring WebSphere Portal to work with DB2 for z/OS.

4. Creating the Java Content Repository database
   View information on setting up your Java Content Repository database to work with WebSphere Portal.

5. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

6. Modifying database properties
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

7. Configuring WebSphere Portal to use DB2 for z/OS
   View information on manually transferring data to the DB2 for z/OS you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository to DB2 for z/OS. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

8. Verifying database connections
   After WebSphere Portal is configured to work with your database, test the database connection to ensure that it operates correctly.

Parent topic: Configuring databases
Installing DB2 for z/OS

View information on how to install DB2 for z/OS for use with IBM® WebSphere® Portal.

The following prerequisites must exist separately from the WebSphere Portal installation:

- If you are using the DB2 Legacy Type 2 JDBC driver, configure your DB2 Connect client with the following commands:
  - `db2 update dbm cfg using tp_mon_name WAS`
  - `db2 update dbm cfg using spm_name hostname`, where `hostname` is the host name for the machine where the DB2 Connect client is installed (same as portal machine).

- If you are using the DB2 Legacy Type 2 JDBC driver, enable extended shared memory usage with the following commands:
  - `export EXTSHM=ON`
  - `db2set DB2ENVLIST=EXTSHM`
  - `db2start`

  To make the change permanent, you must add the environment variable by adding the following to the `profile.env` file:
  ```
  DB2ENVLIST='EXTSHM'
  ```

  in `/home/db2inst/sqllib/userprofile` add: `export EXTSHM=ON`

  **Note:** The shell must be reopened before restarting DB2 for z/OS.

- Both type 2 and type 4 drivers are supported. If you are using the DB2 Legacy Type 2 JDBC driver, the DB2 Connect client must be installed on the same machine as WebSphere Portal to connect to the remote database.

- The DB2 subsystem must be on a supported z/OS platform.

- Update the BP8K0 catalog bufferpool to 35,000 buffers before performing database transfer. You should change this value based on your environment. The SYSIBM.SYSDATABASE table resides in this bufferpool and is used extensively by DB2 for z/OS during database transfer.

To use DB2 for z/OS as the database software for WebSphere Portal, you must have DB2 for z/OS installed on your z/OS system. Refer to the DB2 for z/OS product documentation for instructions. Refer to the Planning for DB2 for z/OS topic for additional considerations for configuring DB2 for z/OS as the WebSphere Portal database.

**Parent topic:** Preparing DB2 for z/OS for a stand-alone production server  
**Next topic:** Creating users
Creating users

Complete this procedure to set up users for IBM® DB2 Universal Database™ for z/OS® to work with IBM WebSphere® Portal.

Before you begin: You should have completed Installing DB2 for z/OS.

- Prerequisites
  - Installing DB2 for z/OS

- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

- If you are configuring multiple WebSphere Portal instances to use a single DB2 for z/OS subsystem, you must create a unique database user for each WebSphere Portal instance. By default all database table names include the name of the database user used to access the data. Therefore, to prevent table name conflicts, you must create a unique database user for each WebSphere Portal instance on the shared DB2 for z/OS subsystem.

- Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Use the following steps to grant access permissions to the database users. Repeat these steps for each WebSphere Portal instance you are setting up.

1. Create all database user IDs in the security product you are using on z/OS. For jcrschema, the database schema name for IBM Java Content Repository data, create a group and connect it to the database user ID for Java Content Repository data, jcr. The following sample shows the RACF definition of such a user ID and group, where jcr is the database user ID for Java Content Repository data, yourDefaultUserGroup is your default RACF group for database user IDs, jcrschema is the database schema name for Java Content Repository data, and yourDefaultGroup is your default RACF group for groups. If you have some other security product such as Top Secret or ACF2 instead of RACF, translate the sample RACF definition into the appropriate syntax before running the job.

```
ADDUSER jcr DFLTGRP(yourDefaultUserGroup) NAME('WAS DB2 ACCESS USER')
PW USER(jcr) NOINTERVAL
ALU jcr PASSWORD(********) NOEXPIRED
ADDGROUP jcrschema SUPGROUP(yourDefaultGroup)
CONNECT jcr GROUP(jcrschema)
```

2. Run the following SQL statements using a tool like SPUFI while logged on to the DB2 subsystem to grant appropriate rights on the newly created databases. If you are configuring multiple WebSphere Portal instances to use a single DB2 for z/OS subsystem, be sure to use the database user associated with the WebSphere Portal instance you are setting up.

```
(C) create/alter tablespaces
(C) create/alter tables
(C) create/alter indice;
(C+R) read/write data
```

(C) - at configuration time

(R) - at runtime
GRANT USE OF ALL BUFFERPOOLS TO releaseusr;
GRANT USE OF ALL BUFFERPOOLS TO communityusr;
GRANT USE OF ALL BUFFERPOOLS TO customizationusr;

GRANT DBADM ON DATABASE jcrdbnameonzos TO jcr;
GRANT USE OF ALL BUFFERPOOLS TO jcr;
GRANT DBADM ON DATABASE fdbkdbnameonzos TO feedback;
GRANT USE OF ALL BUFFERPOOLS TO feedback;
GRANT DBADM ON DATABASE lmdbnameonzos TO lmdbus;
GRANT USE OF ALL BUFFERPOOLS TO lmdbus;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO releaseusr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO communityusr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO customizationusr;

GRANT SELECT ON SYSIBM.SYSTABLES TO releaseusr;
GRANT SELECT ON SYSIBM.SYSTABLES TO communityusr;
GRANT SELECT ON SYSIBM.SYSTABLES TO customizationusr;

GRANT SELECT ON SYSIBM.SYSCOLUMNS TO jcr;
GRANT SELECT ON SYSIBM.SYSTABLES TO jcr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO fdbkdbus;
GRANT SELECT ON SYSIBM.SYSTABLES TO fdbkdbus;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO lmdbus;
GRANT SELECT ON SYSIBM.SYSTABLES TO lmdbus;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO releaseusr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO communityusr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO customizationusr;

GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO jcr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO fdbkdbus;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO lmdbus;
GRANT SELECT ON SYSIBM.SYSSERIES TO releaseusr;
GRANT SELECT ON SYSIBM.SYSSERIES TO communityusr;
GRANT SELECT ON SYSIBM.SYSSERIES TO customizationusr;

GRANT SELECT ON SYSIBM.SYSSERIES TO jcr;
GRANT SELECT ON SYSIBM.SYSSERIES TO fdbkdbus;
GRANT SELECT ON SYSIBM.SYSSERIES TO lmdbus;
GRANT USE OF STOGROUP jcrstogroup TO jcr;
GRANT CREATEIN, DROPIN ON SCHEMA jcrschema TO jcr;
GRANT SELECT ON SYSIBM.SYSTABLESPACE TO jcr;
GRANT SELECT ON SYSIBM.SYSVIEWS TO jcr;
GRANT SELECT ON SYSIBM.SYSDUMMY1 TO jcr;
GRANT SELECT ON SYSIBM.SYSTIGGERS TO jcr;
GRANT SELECT ON SYSIBM.SYSINDEXPART TO jcr;
GRANT SELECT ON SYSIBM.SYSINDEXES TO jcr;
GRANT SELECT ON SYSIBM.SYSSYNONYMS TO jcr;

where:
- releasenameonzos, communinameonzos, customizationnameonzos, and releaseusr, communityusr, customizationusr represent the databases and database users, respectively, of the WebSphere Portal instance you are setting up. (These users must be created on the host system.)
- jcrdbnameonzos and jcr are the database and database user, respectively, for Content Repository data.
- feedbackdbnameonzos and feedback are the database and database user, respectively, for Feedback data.
- lmdbnameonzos and lmdbusr are the database and database user, respectively, for Likeminds data.
- jcrSchema is the database schema name for Content Repository data.

3. Grant the necessary access rights to all users who might require them. Depending on the architecture that you choose, these users might include Java Content Repository and Feedback users.

**Parent topic:** Preparing DB2 for z/OS for a stand-alone production server  
**Previous topic:** Installing DB2 for z/OS  
**Next topic:** Creating remote databases
Creating remote databases

A remote database resides on a different machine than WebSphere Portal. You must manually create the required databases before configuring WebSphere Portal to work with DB2 for z/OS.

Before you begin, complete the following prerequisites:

- These instructions assume that you will use a local IBM® DB2 Universal Database™ Enterprise Server Edition Connect to connect to a remote IBM DB2 Universal Database for z/OS® server, and that IBM WebSphere® Application Server, WebSphere Portal, and DB2 Connect will be installed on the same machine.

- To run SQL statements, you can use a tool like SPUFI. The specified statements include CREATE STOGROUP, CREATE DATABASE and CREATE TABLESPACE. For CREATE STOGROUP you have to replace the `icmvolumes` and `icmvcat` variables with volume serial numbers and a catalog name for your environment. For CREATE TABLESPACE you can specify a specific BUFFERPOOL. Refer to the DB2 for z/OS SQL Reference for a description of these options.

- Ensure that a TEMP database is created for your subsystem. You can use the following statements to create a TEMP database:

  ```sql
  CREATE DATABASE db_name AS TEMP;
  CREATE TABLESPACE ts_name IN db_name;
  ```

  Log on to the DB2 subsystem on the host server. DB2 system administrator rights are needed to create the databases.

  - Replace variables as follows:
    - `releasenameonzos`, `communitynameonzos`, and `customizationnameonzos` are the WebSphere Portal databases for WebSphere Portal and Member Manager data.
    - `fdbkdbnameonzos` and `fdbkdbts` are the database and table space, respectively, for Feedback data.
    - `lmdbnameonzos` and `lmdbts` are the database and table space, respectively, for LikeMinds data.

  - Because large objects are stored in columns that can become very large, logging changes to these columns requires a huge amount of log space. For this reason, large object (LOB) logging is disabled by default for tablespaces that contain such data. With LOB logging disabled, you can recover full backups, but not incremental backups that can be used for point in time recovery. To recover point in time backups, you must enable LOB logging. For detailed instructions, see Technote 1306637, Managing LOB logging in DB2 for z/OS.

- Prerequisites
  - Installing DB2 for z/OS
  - Creating users

Use the following steps to set up WebSphere Portal with a remote instance of DB2 for z/OS. Refer to Planning for DB2 for z/OS for a list of databases and table space names. If you are configuring multiple WebSphere Portal instances to use a single DB2 subsystem, be sure to use the database and table space names associated with the WebSphere Portal instance you are setting up.

1. CREATE DATABASE `releasenameonzos` CCSID UNICODE;
2. CREATE DATABASE `communitynameonzos` CCSID UNICODE;
3. CREATE DATABASE `customizationnameonzos` CCSID UNICODE;
4. Execute the steps in the topic Creating the Java Content Repository database.
5. CREATE DATABASE `fdbkdbnameonzos` CCSID UNICODE;
6. CREATE TABLESPACE `fdbkdbts` IN `fdbkdbnameonzos` USING STOGROUP SYSDEFLT PRIQTY 5000 SECQTY 500 LOCKSIZE ROW;
7. CREATE DATABASE `lmdbnameonzos` CCSID UNICODE;
CREATE TABLESPACE lmdbts IN lmdbnameonzos USING STOGROUP SYSDEFLT PRIQTY 5000 SECQTY 500;

Parent topic: Preparing DB2 for z/OS for a stand-alone production server
Previous topic: Creating users
Next topic: Creating the Java Content Repository database
Creating the Java Content Repository database

View information on setting up your Java Content Repository database to work with WebSphere Portal.

First, you need to create multiple databases for scalability.

The IBM Java Content Repository database grows in size as the IBM® WebSphere® Portal server is used. To support this growth, which includes the dynamic creation of many new tables within the database, the WebSphere Portal server allows you to create multiple databases within a single IBM DB2 Universal Database™ for z/OS® subsystem. A minimum of two databases is recommended, but more can be created if necessary. A single database setup is not recommended; it will quickly become constrained. See the Performance guides on the product documentation Web site for more information.

Each database that is created must begin with a common prefix; there is no convention required for the remainder of the name. For example, to create \( xx \) number of databases, you may choose to use the following commands:

- \( \text{CREATE DATABASE JCRDB01} \)
- \( \text{CREATE DATABASE JCRDB02} \)
- \( \ldots \)
- \( \text{CREATE DATABASE JCRDBxx} \)

In this case, \( \text{JCR} \) is the common prefix. You also have to select a maximum number of user-defined tables (UDT) that each database will be allowed to hold; the recommended value is 400.

- **Prerequisites**
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases

A sample script of the commands that you can use to create the Java Content Repository database in your DB2 for z/OS installation is shown below. The sample demonstrates the creation of a single database. To follow the recommendation of creating at least two databases, duplicate the commands for each additional database as indicated below. Find a template of these commands in the file \( \text{PortalServer_root} \) installer/wp.config/config/templates/db/db2_zos/jcr_sample.sql.

**Notes:**
- It is not necessary to duplicate every tablespace definition in every database.
- In order to run the commands shown in the sample, you must know the database name and the IDs of the MVS volumes where the Java Content Repository database tables will be stored.
- Edit the template file for your installation environment before running the commands shown in the sample:
  - Replace \( \text{jcrdbnameX} \) with the name of the database. Remember that each database name must begin with a common prefix.
  - Replace \( \text{stogroup} \) with the name of your storage group.
  - Replace \( \text{icmvolumes} \) with the IDs of the MVS volumes where the Java Content Repository database tables will be stored. These values are assigned by the database administrator.
  - Replace \( \text{icmvcat} \) with the name of the virtual catalog.
  - Replace \( \text{jcr} \) with the name of database user ID.
  - Replace \( \text{4kbp} \) with the name of your 4K bufferpool.
  - Replace \( \text{32kbp} \) with the name of your 32K bufferpool.
  - Replace \( \text{jcrschema} \) with the schema name of your Java Content Repository domain.
--DROP DATABASE jcrdbnameX
--DROP STOGROUP stogroup
--DROP ALIAS jcrschema.ICMFICTITIOUS

CREATE STOGROUP stogroup VOLUMES (icmvolumes) VCAT icmvcat
GRANT USE OF STOGROUP stogroup to jcr

CREATE DATABASE jcrdbnameX STOGROUP stogroup BUFFERPOOL 4kbp INDEXBP 4kbp CCSID UNICODE

Note: The following two tablespace commands must be created in every database:

CREATE TABLESPACE ICMLFQ32 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICMTSQ04 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

Note: The following tablespaces are required in the first database only:

CREATE TABLESPACE ICMVFQ04 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICMSFQ04 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSTADO IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRIDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRSCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRISLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRGCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRIGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSTDPV IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSDOAC IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RESMLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COLNLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USERLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USEGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE SYSCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSDOAC IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RESMLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COLNLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USEGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE SYSCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5

279
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CACLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CPERLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NLSLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE MAXKLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NLSKLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITTDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITVDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITVILSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COMDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COMALSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COMFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COVDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COVALSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITALLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE IT11LSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE IV11LSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE LI11LSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITTRLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RMATLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CHEOLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE MIMTLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITEELSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
280


LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TSPEJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TSTAJRRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RIDSJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE WSN0JRRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE LOB TABLESPACE ICMLOBT1 IN jcrdbnameX LOG NO
CREATE LOB TABLESPACE ICMLOBT2 IN jcrdbnameX LOG NO
CREATE LOB TABLESPACE ICMLOBT3 IN jcrdbnameX LOG NO
CREATE ALIAS jcrschema.ICMFICTITIOUS FOR SYSIBM.SYSDUMMY1
CREATE TABLESPACE DLKRJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE LKRLJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

Parent topic: Preparing DB2 for z/OS for a stand-alone production server
Previous topic: Creating remote databases
Next topic: Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.
- When using IBM® DB2 Universal Database™ for z/OS® as a data store, WebSphere Portal requires that its indexes are not padded. Therefore, you must set the DSNZPARM parameters to `RETVLCFK=NO` or `PADIX=NO`, or both.

**Prerequisites**
- Installing DB2 for z/OS
- Creating users
- Creating remote databases
- Creating the Java Content Repository database

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`
     For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
     - release
     - community
     - customization
     - jcr
     - feedback
     - likeminds

3. Assign a table space to each `.tablespace` entry in the mapping file. Assignments to `.indexspace` entries are ignored.
   The table space name must be qualified by the database name and prepended by the keyword `IN` and a space. For example: `community.COMP_INST.tablespace=IN COMM8KSPACE`
   Repeat this step for each domain that you are transferring.
4. Save and close `dbdomain.space_mapping.properties`.
5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
- **Windows:** `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
- **UNIX:** `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- **i5/OS:** `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

**Parent topic:** Preparing DB2 for z/OS for a stand-alone production server  
**Previous topic:** Creating the Java Content Repository database  
**Next topic:** Modifying database properties
Modifying database properties
This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases
  - Creating the Java Content Repository database

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

*Tip:* For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:
  - `release.DbName=release`
  - `jcr.DbName=jcrdb`
  - `feedback.DbName=fdbkdb`
  - `likeminds.DbName=lmdb`
  - `community.DbName=commdb`
  - `customization.DbName=custdb`
- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace `dbdomain` with:
  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
  - `dbdomain.DbType`
  - `dbdomain.DbName`
  - `dbdomain.DbUrl`
  - `dbdomain.DbSchema`

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If `DbUser`, `DbUrl`, and `DbPassword` are not the same across domains, the value for `DataSourceName` must differ from the `DataSourceName` of the other domains. In other words, this value must be unique for the database domain.

**Note:** To successfully transfer data from the JCR domain, you must use the DDF location value for the value of `jcr_DbName` field when setting up IBM DB2 Universal Database™ for z/OS®. You can locate the name of the DDF location value in the IBM DB2 Universal Database for z/OS sdsnsamp dataset, member DSNTIJUZ, or by running the following DB2 command:

```
DB2 subsystem prefix display ddf
```

1. Locate the following files and create a backup copy of each before changing any values:
   - `wp_profile_root/ConfigEngine/properties/wkplc.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_comp.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties`

Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In `wkplc_comp.properties`, most properties are repeated for each domain.

2. Use a text editor to open the properties file `wkplc_comp.properties` and modify the values to correspond to your environment.
   
   **A.** For `dbdomain.DbType`, type `db2_zos`.
   
   **B.** For `dbdomain.DbName`, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the `dbdomain.DbUrl` property.
   
   **C.** For `dbdomain.DbSchema`, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   
   **D.** For `dbdomain.DbNameOnZos`, type the name of the WebSphere Portal database on DB2 for z/OS. **Note:**
   - If running DB2 for z/OS as a remote database, set the value to the name of the remote database for the domain.
   - If WebSphere Portal is running on z/OS with DB2 for z/OS, set the value equal to the value of `DbName`.
   
   **E.** For `dbdomain.DataSourceName`, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
   - `releaseDS`
   - `communityDS`
   - `customizationDS`
   - `jcrDS`
   - `lmdbDS`
   - `feedback`

   **F.** For `dbdomain.DbUrl`, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:** The database element of this value should match the value of `DbType`.

287
G. For dbdomain.DbUser, type the user ID for the database administrator.
H. For dbdomain.DbPassword, type the password for the database administrator.
I. For dbdomain.DbTablespace, type the name of the DB2 for z/OS tablespace.
J. For dbdomain.DbStorageGroup, type the name of the storage group for the database.
K. For dbdomain.DbVolumes, type the volumes for the database.
L. For dbdomain.DbVcat, type the VCAT for the database.
M. For dbdomain.Db4KBufferPoolName, type the 4K bufferpool name for the database.
N. For dbdomain.Db32KBufferPoolName, type the 32K bufferpool name for the database.
O. For dbdomain.DbIndex4KBufferPoolName, type the 4K bufferpool name for the database. If you choose to use the default bufferpool value BP3, verify that this bufferpool is active.
P. For dbdomain.TablespaceTrackMod, set the value to determine TRACKMOD attribute of all tablespaces to use the specified value. Refer to the DB2 for z/OS documentation before changing this value.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For source.domainDbType, type of the database you are currently configured to use. The value for source.domainDbType is Derby by default.
B. For source.domainDbName, type the name of the database domain you are currently using.
C. For source.domainDbSchema, type current schema identifier for objects within the database for this domain.
D. For source.domainDataSourceName, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.
E. For source.domainDbUrl, type the url currently used to access your database.
F. For source.domainDbUser, type the name of the user accessing this database.
G. For source.domainDbPassword, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file wkplc_dbtype.properties.

   A. For db2_zos.DbDriver, type the name of the class that SqlProcessor uses to import SQL files.
   B. For db2_zos.DbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.
   C. For db2_zos.JdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.
   D. For db2_zos.DbDriverType, type the number of the driver type for the database.
   E. For db2_zos.DbLocationName, type the DB2 location name. This value is set in the installation job DSNTIJUZ.

6. Save and close the file.

7. Update the following property in the file wkplc.properties.

   A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Preparing DB2 for z/OS for a stand-alone production server

**Previous topic:** Assigning custom table spaces

**Next topic:** Configuring WebSphere Portal to use DB2 for z/OS
Configuring WebSphere Portal to use DB2 for z/OS

View information on manually transferring data to the DB2 for z/OS you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository to DB2 for z/OS. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

- Prerequisites
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases
  - Creating the Java Content Repository database
  - Modifying database properties

Steps for transferring data to another supported database

1. If you are running a type 2 connection, before transferring data edit the db2cl.ini file. Failure to follow these steps will cause the database transfer to hang at the task action-process-constraints.
   A. Locate the file /home/db2inst1/sqllib/cfg/db2cl.ini.
   B. Add the following to the end of the file. Leave an empty line after ReturnAliases=0.[COMMON]

   DYNAMIC=1
   ReturnAliases=0

2. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.
3. Enter the following commands to validate configuration properties ...

   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password

4. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td>./stopServer.sh server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td>./stopServer.sh WebSphere_Portal - username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

6. Transfer the database:
   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.
   A. Change to the directory wp_profile_root/ConfigEngine.
B. Enter the following command:

```bash
./ConfigEngine.sh database-transfer -DWasPassword=password
```

**Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:

```bash
./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
```

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

7. From the command line processor, reset the check pending status on the WebSphere Portal table spaces listed here. CHECK DATA is a DB2 batch utility that is invoked through JCL or through the DB2 utility panels. Type the following utility commands to check pending status:

```bash
check data tablespace releasenameonzos.TS280A
check data tablespace releasenameonzos.TS300A
check data tablespace releasenameonzos.TS2110A
check data tablespace releasenameonzos.TS830A
check data tablespace communinameonzos.TS8000B
check data tablespace communinameonzos.TS8011B
check data tablespace communinameonzos.TS280B
check data tablespace customizationnameonzos.TS2110C
check data tablespace jcrdbnameonzos.ICMSFQ04
check data tablespace jcrdbnameonzos.TS2110D
```

where `releasenameonzos`, `communinameonzos`, and `customizationnameonzos` are the names of your WebSphere Portal databases, and `jcrdbnameonzos` is the name of your JCR database. Refer to the Utility Guide and Reference for DB2 for z/OS for additional details.

8. Run the **RUNSTATS** utility as shown in the following example:

```bash
LISTDEF JCRDBZOS INCLUDE TABLESPACE JCRDBZOS.* BASE
RUNSTATS TABLESPACE LIST JCRDBZOS INDEX(ALL) KEYCARD TABLE(ALL)
LISTDEF JCRDB01 INCLUDE TABLESPACE JCRDB01.* BASE
RUNSTATS TABLESPACE LIST JCRDB01 INDEX(ALL) KEYCARD TABLE(ALL)
...
```

9. Start the WebSphere Application Server.

10. Verify that the WebSphere Portal application server is running by opening the following URL in a browser:

```
http://hostname.companyname.com:port_number/wps/portal
```

where `hostname.companyname.com` is the fully qualified host name of the machine where WebSphere Portal is running and `port_number` is the transport port that is created by WebSphere Application Server.

If you have performed database transfer in a clustered environment, you must explicitly copy the `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm/icm.properties` file from the primary node on which the `database-transfer` task was processed to all secondary nodes. This ensures that the secondary nodes have the new JCR database values specified in the `icm.properties` file.

1. Stop the portal server on the secondary node.
2. Copy the `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm/icm.properties` file from the primary node and replace the `icm.properties` file on the secondary node with the new file from the primary node.
3. Start the portal server on the secondary node.

**Parent topic:** Preparing DB2 for z/OS for a stand-alone production server

**Previous topic:** Modifying database properties
Verifying database connections

After WebSphere Portal is configured to work with your database, test the database connection to ensure that it operates correctly.

**Prerequisites**
- Installing DB2 for z/OS
- Creating users
- Creating remote databases
- Creating the Java Content Repository database
- Modifying database properties
- Configuring WebSphere Portal to use DB2 for z/OS

You can verify the connection from a browser or from a command line. To verify that WebSphere Portal is running from a browser, open the portal in a Web browser: http://hostname.yourco.com:port_number/wps/portal, where hostname.yourco.com is the fully qualified host name of the machine where WebSphere Portal is running and port_number is the transport port that is created by IBM® WebSphere® Application Server.

There may be an error if any of the following conditions appear.
- When trying to access the portal you get a 503 error.
- If you had any locale problems with your database, you could see invalid characters, such as ????, after logging in. This may happen if the character set of the database is not UTF-8 compliant.
- If something went wrong with the data that was transferred, you may not be able to login. WebSphere Portal will indicate you entered an invalid user ID and password even though you know it is valid.

Verify the connection from a command line by completing the following steps:

1. Start a 5250 session on the local machine where WebSphere Portal is installed.
2. For WebSphere Portal on WebSphere Application Server (UserData path), enter the following on the command line: cd wp_profile_root/ConfigEngine.
3. Enter the following command: `ConfigEngine.sh validate-database-connection -DTransferDomainList=release,community,customization,jcr,feedback,likeminds -DWasPassword=password`

For security reasons, you should not leave passwords in the wkplc_comp.properties file. Edit the file prior to running a configuration task and insert the passwords that are needed for that task. After the task has run, delete all passwords from the file.

Alternatively, you can specify the password on the command line rather than update the wkplc_comp.properties file. For example: `ConfigEngine.sh -DPortalAdminPwd=password -DWasPassword=password validate-wps-admin-login`

When installing WebSphere Portal, the passwords in the wkplc_comp.properties file are automatically removed after configuration.

**Parent topic:** Preparing DB2 for z/OS for a stand-alone production server

**Previous topic:** Configuring WebSphere Portal to use DB2 for z/OS
Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Setting up the Oracle database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

1. **Installing Oracle**
   View information on how to install Oracle for use with WebSphere Portal.

2. **Creating databases**
   View some important considerations before setting up Oracle databases to work with WebSphere Portal.

3. **Creating users**
   View the steps to set up users for Oracle to work with WebSphere Portal.

4. **Modifying database properties**
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. **Setting up databases**
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

6. **AIX stand-alone: Creating JCR table spaces**
   This topic provides manual instructions for creating JCR table spaces for Oracle.

7. **Optional: Assigning custom table spaces**
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

8. **Configure WebSphere Portal to use Oracle**
   This section provides information on how to manually transfer data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

**Parent topic:** Configuring databases
Installing Oracle

View information on how to install Oracle for use with WebSphere Portal.

You can use Oracle as the database software. You must install Oracle separately and before performing a database transfer.

Refer to Oracle product documentation for installation instructions.

---

**Parent topic:** Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Next topic:** Creating databases
Creating databases

View some important considerations before setting up Oracle databases to work with WebSphere Portal.

For information about creating databases, refer to the Oracle product documentation. For information on the recommended database architecture and the databases you will need to create, see the Planning for Oracle topic. Be sure that all databases to be used with WebSphere Portal are created as UNICODE character set databases.

If you are using remote Oracle 9 or 10 databases, you must also copy the ojdbc14.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM® WebSphere® Application Server Version 7.0, you must also copy the ojdbc6.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM WebSphere Application Server Version 6.1, you must also copy the ojdbc5.jar file from the remote Oracle server to the WebSphere Portal machine. The typical location is the oracle_home/sqldeveloper/jdbc/lib directory. Record the copy location on your local machine for future reference.

When creating Oracle databases for use with WebSphere Portal, you should consider the following information:

- All databases must be created manually before configuring WebSphere Portal.
- All databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle server will be remote or local to the WebSphere Portal installation.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle database in order for WebSphere Portal to communicate with the Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools.

  **Recommended initial buffer pool sizes:**
  
  - `db_block_size = 8192` bytes
  - `db_cache_size = 307,200` bytes
  - `db_files = 1024` files
  - `log_buffer = 65536` bytes
  - `open_cursors = 1500` cursors
  - `pga_aggregate_target = 204,800` bytes
  - `pre_page_sga = true`
  - `processes = 300` processes
  - `shared_pool_size = 204,800` bytes

  **Note:** If you are using IBM Java Content Repository, the `open_cursors` value may need to be increased based on the table count in the Java Content Repository schema.

- Raise the number of parallel servers as appropriate. For example, if you have more than 875 parallel servers, you should set the `parallel_max_servers` to 1200.

- The Oracle parameter CURSOR_SHARING allows similar SQL Statements to be shared when possible, which prevents parsing and establishing a new execution plan. The execution plan is used by Oracle to gather the data needed to satisfy a request. There are two options for CURSOR_SHARING, which are as follows:

  - **FORCE**
    - When you select this option, Oracle uses the same execution plan for all SQLs that are similar in value even if the values are different. When you use this option, the execution plan may not provide optimum performance. For example, similar SQLs with different values may behave differently when executed running the same plan.

  - **EXACT**
- When you select this option, Oracle only shares the same execution plan for SQLs that are identical and use the same values. This option removes the risk of a SQL statement being executed when optimum performance conditions do not exist.

WebSphere Portal supports both options. Regardless of the option selected, portlet applications should not be affected. Contact your database administrator for further assistance on these options.

- **Prerequisites**
  - Installing Oracle

**Parent topic:** Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Installing Oracle

**Next topic:** Creating users
Creating users

View the steps to set up users for Oracle to work with WebSphere Portal.

Before you begin:

- You should have installed Oracle.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

**Prerequisites**
- Installing Oracle
- Creating databases

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Start the Oracle SQL command line processor ‘SQL*Plus’ by entering `sqlplus /nolog` on the operating system command prompt.
2. Log in to the Oracle database by executing command `connect user/password@database name`. For example, `system/manager@wpsdb` will log the administrative user system with a password of manager into the wpsdb database.
3. Create the WebSphere Portal user `dbdomain.DbUser`, where `dbdomain` is replaced by release, community, or customization.
   A. Log in to the database in which you want to create the new users.
   B. Create the users that you need in the appropriate databases by entering the following commands in the SQL*Plus tool, where `user_tablespace` is the default tablespace that is identified by the database administrator to store user objects and `temp_tablespace` is identified to store temporary objects. All names must be in lower case.

   ```sql
   SQL> create user releaseusr identified by password
   default tablespace user_tablespace
   temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO releaseusr;
   SQL> create user communityusr identified by password
   default tablespace user_tablespace temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO communityusr;
   SQL> create user customizationusr identified by password
   default tablespace user_tablespace
   temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO customizationusr;
   SQL> create user jcr identified by password
   default tablespace user_tablespace
   temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO jcr
   ```
4. Connect to the Feedback database:
   A. Enter the following command: `SQL> connect`
B. Enter user-name: username/password@dbname where username is an existing administrative user in the database. For example, system/manager@fdbkdb will log the administrative user system with a password of manager into the fdbkdb database.

C. Create the Feedback user:

```sql
SQL> create user feedback identified by password
    
default tablespace user_tablespace
    
temporary tablespace temp_tablespace;
    
GRANT UNLIMITED TABLESPACE TO feedback
```

D. Log out of the command line tool using the command SQL> exit.

5. Connect to the LikeMinds database:

A. Enter the following command:

```sql
SQL> connect
```

B. Enter user-name: username/password@dbname, where username is an existing administrative user in the database.

For example, system/manager@lmdb will log the administrative user system with a password of manager into the lmdb database.

C. Create the LikeMinds user:

```sql
SQL> create user lmdbusr identified by password
    
default tablespace user_tablespace
    
temporary tablespace temp_tablespace;
    
GRANT UNLIMITED TABLESPACE TO lmdbusr
```

D. Log out of the command line tool using the command SQL> exit.

6. Perform the following steps to verify that you can connect to the database with the user ID and password that you just created:

A. Enter the SQL> connect command to connect to the content database.

B. Enter user-name: username/password@dbname, where username is an existing administrative user in the database. For example, system/manager@jcrdb will log the administrative user system with a password of manager into the Oracle database.

---

**Parent topic:** Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server  
**Previous topic:** Creating databases  
**Next topic:** Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Installing Oracle
  - Creating databases
  - Creating users

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:
  - `release.DbName=release`
  - `jcr.DbName=jcrdb`
  - `feedback.DbName=fdbkdb`
  - `likeminds.DbName=lmdb`
  - `community.DbName=commdb`
  - `customization.DbName=custdb`

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace `dbdomain` with:
  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`

- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
- \texttt{dbdomain.DbType}
- \texttt{dbdomain.DbName}
- \texttt{dbdomain.DbUrl}
- \texttt{dbdomain.DbSchema}

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If \texttt{DbUser}, \texttt{DbUrl}, and \texttt{DbPassword} are not the same across domains, the value for \texttt{DataSourceName} must differ from the \texttt{DataSourceName} of the other domains. In other words, this value must be unique for the database domain.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and Like Minds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

1. Locate the following files and create a backup copy of each before changing any values:
   - \texttt{wp_profile_root}/ConfigEngine/properties/wkplc.properties
   - \texttt{wp_profile_root}/ConfigEngine/properties/wkplc_comp.properties
   - \texttt{wp_profile_root}/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In \texttt{wkplc_comp.properties}, most properties are repeated for each domain.

2. Use a text editor to open the properties file \texttt{wkplc_comp.properties} and modify the values to correspond to your environment.
   A. For \texttt{dbdomain.DbType}, type \texttt{oracle}.
   B. For \texttt{dbdomain.DbName}, type the name of the WebSphere Portal domain database. \textbf{Note:} This value is also the database element in the \texttt{dbdomain.DbUrl} property.
   C. For \texttt{dbdomain.DbSchema}, type the schema name of the database domain. \textbf{Note:} Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The \texttt{dbdomain.DbName} should be the same value used for the \texttt{dbdomain.DbSchema}
      \textbf{Restriction:} The value for \texttt{dbdomain.DbSchema} must equal the value for \texttt{dbdomain.DbUser}.
   D. For \texttt{dbdomain.DataSourceName}, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For \texttt{dbdomain.DbUrl}, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. \textbf{Note:}
      - The database element of this value should match the value of \texttt{DbName}.
      - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following:
        \texttt{jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME}. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.
F. For $dbname$.DbUser, type the user ID for the database administrator. **Restriction:** The value for $dbname$.DbUser must equal the value for $dbname$.DbSchema.

G. For $dbname$.DbPassword, type the password for the database administrator.

H. For $dbname$.DBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.

I. For $dbname$.DBA.DbPassword, type the database administrator password for privileged access operations during creation of the database.

J. For $dbname$.DbHome, type the root location for the database. **Note:** This value is used to specify the location to create the tables.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

   A. For source.domainDbType, type of the database you are currently configured to use. The value for source.domainDbType is Derby by default.

   B. For source.domainDbName, type the name of the database domain you are currently using.

   C. For source.domainDbSchema, type current schema identifier for objects within the database for this domain.

   D. For source.domainDataSourceName, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

   E. For source.domainDbUrl, type the url currently used to access your database.

   F. For source.domainDbUser, type the name of the user accessing this database.

   G. For source.domainDbPassword, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file wkplc_dbtype.properties.

   A. For oracle.DbDriver, type the name of the class that SqlProcessor uses to import SQL files.

   B. For oracle.DbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For oracle.JdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file wkplc.properties.

   A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Creating users

**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

- Prerequisites
  - Installing Oracle
  - Creating databases
  - Creating users
  - Modifying database properties

1. On the database server, make sure that the folder `db_1` contains the subfolders `your_oracle_instance/data` and `your_oracle_instance/index`. If this folder hierarchy does not exist, create it manually before you run the `setup-database` task. The `setup-database` task requires these folders to create database users. If these folders do not exist, the `setup-database` task will fail.

2. Change to the directory `wp_profile_root/ConfigEngine`

3. To create the database users, type the following command:

   ```bash
   ./ConfigEngine.sh setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:

   ```sql
   '<WP_root>/base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql'.
   ```

Parent topic: Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Modifying database properties

Next topic: AIX stand-alone: Creating JCR table spaces

Related tasks
Manually creating users and granting privileges for Oracle
AIX stand-alone: Creating JCR table spaces

This topic provides manual instructions for creating JCR table spaces for Oracle.

- **Prerequisites**
  - Installing Oracle
  - Creating databases
  - Creating users
  - Modifying database properties
  - Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.

2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:
   - `&jcrdb` is the name of the database you created to store user data.
   - `&dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.

   Ensure that the `.` is included in the variables when you substitute the values of your environment with these variables.

   **Important:** You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

```sql
create tablespace ICMLFQ32
datafile 'datafile.&dbpath./&jcrdb._ICMLFQ32_01.dbf'
sizE300M reuse autoextend on
next 10M maxsize UNLIMITED extent management local autoallocate;
create tablespace ICMLNF32
datafile 'datafile.&dbpath./&jcrdb._ICMLNF32_01.dbf'
sizE25M reuse autoextend on
next 10M maxsize UNLIMITED extent management local autoallocate;
create tablespace ICMVFQ04
datafile 'datafile.&dbpath./&jcrdb._ICMVFQ04_01.dbf'
sizE25M reuse autoextend on
next 10M maxsize UNLIMITED extent management local autoallocate;
create tablespace ICMSFQ04
datafile 'datafile.&dbpath./&jcrdb._ICMSFQ04_01.dbf'
sizE150M reuse autoextend on
next 10M maxsize UNLIMITED extent management local autoallocate;
create tablespace ICMLSNDX
datafile 'indexfile.&dbpath./index/.ICMLSNDX_01.dbf'
sizE10M reuse autoextend on
next 10M maxsize UNLIMITED extent management local autoallocate;
```

A. Set the `size`, `autoextend`, and `maxsize` values according to your environment. For example, you may want to change the `maxsize` to a set value rather than `UNLIMITED`.

B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.

C. Refer to the Oracle command reference for more information about using the `create tablespace` command.

**Parent topic:** Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Setting up databases

**Next topic:** Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the wp_profile_root/PortalServer/config/tables/tables/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

- Prerequisites
  - Installing Oracle
  - Creating databases
  - Creating users
  - Modifying database properties
  - Setting up databases
  - AIX stand-alone: Creating JCR table spaces

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file wp_profile_root/PortalServer/config/tables/tables/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.index_name.indexspace
   - For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
     - release
     - community
     - customization
     - jcr
     - feedback
     - likeminds
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword TABLESPACE and a space. For example: community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE
   - Repeat this step for each domain that you are transferring.
4. Save and close dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
- **UNIX**: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- **i5/OS**: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

**Parent topic:** Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** AIX stand-alone: Creating JCR table spaces

**Next topic:** Configure WebSphere Portal to use Oracle
Configure WebSphere Portal to use Oracle

This section provides information on how to manually transfer data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- Prerequisites
  - Installing Oracle
  - Creating databases
  - Creating users
  - Modifying database properties
  - Setting up databases
  - AIX stand-alone: Creating JCR table spaces

Tips:
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

Steps for transferring data to another supported database

1. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.
2. Enter the following commands to validate configuration properties:
   ```
   ./%ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ./%ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```
3. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.
4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td>./stopServer.sh server1 -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>
5. Transfer the database. **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

A. Change to the directory `wp_profile_root/ConfigEngine`.

B. Enter the following command:

```
./ConfigEngine.sh database-transfer -DWasPassword=password
```

**Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:

```
./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
```

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example:

```
SQL> execute dbms_stats.gather_schema_stats(ownname=> 'jcr', cascade=> TRUE);
```

7. Change to the directory `wp_profile_root/bin`.

8. Enter the following command to start the WebSphere Portal server:

```
./startServer.sh WebSphere_Portal
```

---

**Parent topic:** Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Assigning custom table spaces
Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Setting up the Oracle RAC database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

1. **Installing Oracle RAC**
   - View information on how to install Oracle RAC for use with WebSphere Portal.

2. **Create users**
   - View the steps to set up users for Oracle RAC to work with WebSphere Portal.

3. **Creating databases**
   - View some important considerations before setting up Oracle RAC databases to work with WebSphere Portal.

4. **Modifying database properties**
   - This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. **Setting up databases**
   - This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

6. **Optional: Assigning custom table spaces**
   - The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

7. **Configuring WebSphere Portal to use Oracle RAC**
   - This section provides information on how to manually transfer data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

**Parent topic:** Configuring databases
Installing Oracle RAC

View information on how to install Oracle RAC for use with WebSphere Portal.

Before you begin:
- You should have completed reviewing the Planning for Oracle RAC section.
- Ensure the Oracle CRS (Cluster Ready Service) and Oracle RAC databases have been installed and configured on the primary and secondary nodes.
- Run the following commands on both RAC nodes to start global services daemon_(GSD), oracle listeners, and agents.
  - `$ gsdctl start`
  - `$ lsnrctl start`
  - `$ agentctl start`

You can use Oracle RAC as the database software for WebSphere Portal. You must install Oracle RAC separately and before performing WebSphere Portal database transfer.

Refer to Oracle product documentation for installation instructions.

Parent topic: Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Next topic: Create users
Create users

View the steps to set up users for Oracle RAC to work with WebSphere Portal.

Before you begin:
- You should have completed installing Oracle RAC.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for
  WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

**Prerequisites**

- Installing Oracle RAC

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Create the users in the WebSphere Portal databases
   A. Log in to the database in which you want to create the new users.
   B. Create the users that you need in the appropriate databases by entering the following command in the SQL*Plus tool, where `user_tablespace` is the default tablespace that is identified by the database administrator to store user objects and `temp_tablespace` is identified to store temporary objects: **Tip:** Balance the storage of user objects among tablespaces to prevent running out of space with overuse of `user_tablespace`. Also consider increasing the size of `user_tablespace` when handling a high volume of users.

   ```sql
   SQL> create user username identified by password
   default tablespace user_tablespace
   temporary tablespace temp_tablespace;
   ```

2. Log in by entering the command `sqlplus` in SQL*Plus:

3. Enter the command `username: username/password@dbname`, where `username` is an existing administrative user in the database. For example: `system/manager@wpsdb` will log the administrative user system with a password of manager into the `wpsdb` database.

4. Create the WebSphere Portal user `dbdomain.DbUser`, where `dbdomain` is replaced by release, community, customization, or JCR. All names must be in lower case. The following code uses the default user tablespace and the default temporary tablespace. Explicitly specify these tablespaces if you do not want to use the defaults.

   ```sql
   SQL> create user releaseusr identified by password;
   SQL> create user communityusr identified by password;
   SQL> create user customizationusr identified by password;
   SQL> create user jcr identified by password;
   ```

5. Connect to the content database by entering the command `sqlplus` in SQL*Plus:

6. Enter the following, where `username` is an existing administrative user in the database: `username: username/password@dbname` For example, `system/manager@jcrdb` will log the administrative user system with a password of manager into the Oracle RAC database.

7. Create the Oracle RAC users by entering the following: When creating the `jcr` user, grant all necessary privileges. If you do not grant privileges, you will receive the error `ICMADMIN lacks CREATE SESSION Privilege logon denied`
when you try to connect with the `jcr` user:

```
SQL> create user jcr identified by password default tablespace users temporary tablespace temp;
```

8. Connect to the Feedback database:
   A. Enter the following command: `SQL> connect`  
   B. Enter `user-name: username/password@dbname` where `username` is an existing administrative user in the database. For example, `system/manager@fdbkdb` will log the administrative user system with a password of manager into the `fdbkdb` database.

9. Create the Feedback user:
   ```
   SQL> create user feedback identified by password default tablespace users temporary tablespace temp;
   ```

10. Connect to the LikeMinds database:
```
    SQL> connect
    ```
   11. Enter `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example, `system/manager@lmdb` will log the administrative user system with a password of manager into the `lmdb` database.

12. Create the LikeMinds user:
```
    SQL> create user lmdbusr identified by password default tablespace users temporary tablespace temp;
    ```

13. Log out of the command line tool using the command `SQL> exit`.

Parent topic: Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server  
Previous topic: Installing Oracle RAC  
Next topic: Creating databases
Creating databases

View some important considerations before setting up Oracle RAC databases to work with WebSphere Portal. When creating Oracle RAC databases for use with WebSphere Portal, you should consider the following information:

- The Oracle RAC databases must be created manually before configuring WebSphere Portal.
- All databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle RAC server will be remote or local to the WebSphere Portal installation.
- If using an earlier version of Oracle RAC (9i or earlier), ensure that Oracle RAC JVM is also installed.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle RAC database in order for WebSphere Portal to communicate with the IBM Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes are:

  - db_block_size = 8192 bytes
  - db_cache_size = 314,572,800 bytes
  - db_files = 1024 files
  - log_buffer = 65536 bytes
  - open_cursors = 1500 cursors
  - pga_aggregate_target = 209,715,200 bytes
  - pre_page_sga = true
  - processes = 300 processes
  - shared_pool_size = 209,715,200 bytes

Note: If you are using IBM Java Content Repository, the `open_cursors` value may need to be increased based on the table count in the IBM Java Content Repository schema.

- Prerequisites
  - Installing Oracle RAC
  - Create users

Refer to the following instructions to create tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.
2. Create the tablespaces. Important: You must use the same tablespace names listed in the commands. The tablespace names cannot be customized or modified.
   
   A. Find and edit the SQL script `jcr_ora_tablespaces.sql` in the directory `wp_profile_root/ConfigEngine/work/db/oracle`.
   
   B. In the define section, replace the following variables with the values from your environment:

      - `jcrdb`
        - Name of the database you created to store user data.
      - `logfile`
        - Location to store the log file.
      - `dbpath`
        - Directory where you created the database.
   
   C. Set the size, autoextend, and maxsize values according to your environment. For example, you may want to change the maxsize to a set value rather than UNLIMITED. Consult your Database Administrator for more info.
D. Execute the following SQL script:

**Notes:**
- The DBA or a user with sufficient credentials (CREATE tablespace) must execute the script.
- The script will prompt for the database username and password.

```sql
# sqlplus
SQL > @jcr_ora_tablespaces.sql
```

Parent topic: Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Create users

Next topic: Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- Prerequisites
  - Installing Oracle RAC
  - Create users
  - Creating databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

Tip: For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Working with properties files:
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:
  - release.DbName = release
  - jcr.DbName = jcrdb
  - feedback.DbName = fdbkdb
  - likeminds.DbName = lmdb
  - community.DbName = commdb
  - customization.DbName = custdb

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds

- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
For **dbdomain.DbType**, type `oracle`.

**dbdomain.DbName**

- For **dbdomain.DbUser**, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the `dbdomain.DbUrl` property.

- For **dbdomain.DbName**, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the `dbdomain.DbUrl` property.

- For **dbdomain.DbSchema**, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The `dbdomain.DbName` should be the same value used for the `dbdomain.DbSchema`.

  **Restriction:** The value for `dbdomain.DbSchema` must equal the value for `dbdomain.DbUser`.

- For **dbdomain.DataSourceName**, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
  - `releaseDS`
  - `communityDS`
  - `customizationDS`
  - `jcrDS`
  - `lmdbDS`
  - `feedback`

- For **dbdomain.DbUrl**, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:**
  - The database element of this value should match the value of `DbName`.
  - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following:

    `jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANECENAME`. **When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.**
F. For `dbdomain.DbUser`, type the user ID for the database administrator. **Restriction:** The value for `dbdomain.DbUser` must equal the value for `dbdomain.DbSchema`.

G. For `dbdomain.DbPassword`, type the password for the database administrator.

H. For `dbdomain.DBA.DbUser`, type the database administrator user ID for privileged access operations during creation of the database.

I. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.

J. For `dbdomain.DbHome`, type the root location for the database. **Note:** This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

   A. For `source.domainDbType`, type of the database you are currently configured to use. The value for `source.domainDbType` is Derby by default.

   B. For `source.domainDbName`, type the name of the database domain you are currently using.

   C. For `source.domainDbSchema`, type current schema identifier for objects within the database for this domain.

   D. For `source.domainDataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

   E. For `source.domainDbName`, type the url currently used to access your database.

   F. For `sourcedomain.DbUser`, type the name of the user accessing this database.

   G. For `source.domainDbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

   A. For `oracle.DbDriver`, type the name of the class that `SqlProcessor` uses to import SQL files.

   B. For `oracle.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For `oracle.jdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Creating databases

**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

Before you begin, ensure that the following prerequisites are met:
- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

**Prerequisites**

- Installing Oracle RAC
- Create users
- Creating databases
- Modifying database properties

1. On the database server, make sure that the folder `db_1` contains the subfolders `your_oracle_instance/data` and `your_oracle_instance/index`. If this folder hierarchy does not exist, create it manually before you run the `setup-database` task. The `setup-database` task requires these folders to create database users. If these folders do not exist, the `setup-database` task will fail.

2. Change to the directory `wp_profile_root/ConfigEngine`

3. To create the database users, type the following command:

   ```bash
   ./ConfigEngine.sh setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:

   ```
   '<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql'.
   ```

**Creating JCR table spaces**

This topic provides manual instructions for creating JCR table spaces for Oracle.

**Parent topic:** Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Modifying database properties

**Next topic:** Assigning custom table spaces

**Related tasks**

Manually creating users and granting privileges for Oracle
Creating JCR table spaces

This topic provides manual instructions for creating JCR table spaces for Oracle. Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.
2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:
   - `&jcrdb` is the name of the database you created to store user data.
   - `&dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.
   Ensure that the `.` is included in the variables when you substitute the values of your environment with these variables.

   **Important:** You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

   ```
   create tablespace ICMLFQ32 datafile '&dbpath./&jcrdb./data/ICMLFQ32_01.dbf' size 300M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLNF32 datafile '&dbpath./&jcrdb./data/ICMLNF32_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLVFQ04 datafile '&dbpath./&jcrdb./data/ICMLVFQ04_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLSNDX datafile '&dbpath./&jcrdb./index/ICMLSNDX_01.dbf' size 10M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   ```

   A. Set the `size`, `autoextend`, and `maxsize` values according to your environment. For example, you may want to change the `maxsize` to a set value rather than `UNLIMITED`.

   B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.

   C. Refer to the Oracle command reference for more information about using the `create tablespace` command.

**Parent topic:** Setting up databases
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:

- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root` /PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

**Prerequisites**

- Installing Oracle RAC
- Create users
- Creating databases
- Modifying database properties
- Setting up databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`
   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds

3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword `TABLESPACE` and a space. For example: `community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE`
   Repeat this step for each domain that you are transferring.

4. Save and close `dbdomain.space_mapping.properties`.

5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example, for Windows:
   ```
   Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
   ```
- UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

Parent topic: Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Setting up databases

Next topic: Configuring WebSphere Portal to use Oracle RAC
Configuring WebSphere Portal to use Oracle RAC

This section provides information on how to manually transfer data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- Prerequisites
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties
  - Setting up databases

Tips:

- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

- The WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following: jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.

- Manually specify an RAC datasource URL in the WebSphere Application Server console. The URL should look like the following:

  jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT=1521)))(ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICENAME)).

- When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

Steps for transferring data to another supported database

1. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.
2. Enter the following commands to validate configuration properties.

```
./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
```

```
./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
```

3. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.

4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td><code>./stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td><code>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

5. Transfer the database: 

   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory `wp_profile_root`/ConfigEngine.

   B. Enter the following command: 

```
./ConfigEngine.sh database-transfer -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
```

   **Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command: 

```
./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
```

   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example:

```
SQL> execute dbms_stats.gather_schema_stats(ownname=> 'jcr', cascade=> TRUE);
```

7. Specify the JDBC URL to connect to the cluster:

   A. Login to the WebSphere Application Server Administrator Console

   B. Navigate to Resources > JDBC Providers

   C. If there is a value in the Node field, remove it and click Apply.

   D. For each Oracle JDBC provider, repeat the following steps:

   1. Click the provider name.

   2. Select Data Sources.

   3. Click the name of the data source.

   4. Under the Oracle data source properties, modify the URL attribute according to Oracle RAC configuration:

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT=1521)) (ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICENAME)))
```

   5. Save your changes

8. Change to the directory `wp_profile_root/bin`.

9. Enter the following command to start the WebSphere Portal server:

```
./startServer.sh WebSphere_Portal
```

---

**Parent topic:** Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Assigning custom table spaces
Setting up a remote SQL Server database on Windows for a stand-alone production server

To setup a remote Microsoft SQL Server Enterprise Edition database, create user IDs and databases on a remote server. Tasks are provided to assist with creating the user IDs and the databases. Before you can use the tasks, you must modify properties files.

1. Installing SQL Server
   View the steps to install SQL Server for use with WebSphere Portal.

2. Modifying database properties
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

3. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

   The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

5. Configuring WebSphere Portal to use SQL Server 2005
   View information on manually transferring data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configuring databases
Installing SQL Server

View the steps to install SQL Server for use with WebSphere Portal.

Before you begin this task, complete the following prerequisites:
- You should have completed reviewing the Planning for SQL Server topic.
- You must install SQL Server separately from WebSphere Portal.
- WebSphere Portal requires either the DataDirect Connect for JDBC 3.7 JDBC driver or the Microsoft SQL Server JDBC Driver.
- You can obtain the DataDirect Driver from DataDirect Technologies. The JDBC driver consists of two parts: the JDBC driver libraries and JTA libraries. The JDBC driver libraries must be put on the WebSphere Portal system, while the JTA libraries are intended for SQL Server.
- You can obtain the Microsoft SQL Server JDBC Driver from Microsoft. See the Microsoft SQL Server product documentation for installation details.

The following section provides instructions for installing SQL Server for use with IBM® WebSphere® Application Server and WebSphere Portal. These steps are the same for both the DataDirect and Microsoft drivers unless noted.

1. Install SQL Server and all required patches.
2. Select the authentication mode for this installation:
   - Windows Authentication Mode
   - Mixed Mode (Windows Authentication and SQL Server Authentication)
     Either authentication mode allows the user who is specified as the database administrator in the \wkplc_comp.properties file to log in to the SQL server. Mixed Mode authentication allows either a Windows user or a SQL server user, or both, to log in to the SQL Server.
3. In the SQL Server Setup panel, **Components to Install**, select the following components, which are required services for WebSphere Portal:
   - **SQL Server Database Services**
   - **Integration Services**
     The option **Integration Services**, creates the directory DTS/Binn, where you will need to copy files when you install Microsoft SQL Server JDBC drivers and enable XA connections.
4. Complete the installation using SQL Server documentation as a guide.
5. Enable TCP/IP connectivity in the **SQL Server Configuration Manager**.
6. Install the JDBC driver using one of these methods:
   - Installing DataDirect Connect for JDBC drivers on UNIX
   - Installing DataDirect Connect for JDBC drivers on Windows
   - Installing Microsoft SQL Server JDBC drivers and enabling XA connections
7. Network the connections for MSDTC in the Windows Component Services as described in Microsoft Help and Support.
8. Start SQL Server.

**Installing DataDirect Connect for JDBC drivers on UNIX**

1. Purchase and download DataDirect Connect for JDBC and save file 360connectjdbc.jar in a temporary work directory.
2. To create the required files, run the following command from the directory that contains 360connectjdbc.jar:

```
jar -xvf 360connectjdbc.jar
```

3. Run `./Installer.sh` in the same directory.

4. When installation is complete, change directory to `/installation/lib`, where `installation` is the directory in which you installed the DataDirect Connect for JDBC drivers.

5. Run the following command to change the permissions on the JAR files in the `installation/lib` directory:

```
chmod 777 *.jar
```

6. Change the ownership and group of these files by running the following commands in the `installation/lib` directory:

```
chgrp system_grp *.jar
chown root *.jar
```

   Where `system_grp` is the system group as labeled by your operating system.

---

### Installing DataDirect Connect for JDBC drivers on Windows

1. Purchase and download DataDirect Connect for JDBC and save file `360connectjdbc.jar` in a temporary work directory.

2. To create the required files, run the following command from the directory that contains `360connectjdbc.jar`:

```
jar -xvf 360connectjdbc.jar
```

3. Run `Installer.bat` in the same directory.

---

### Installing Microsoft SQL Server JDBC drivers and enabling XA connections

1. Download and install the Microsoft SQL Server JDBC driver.

2. Copy file `sqljdbc_xa.dll` from the `xa` subdirectory to the `C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqljdbc_xa.dll` directory of the SQL Server installation.

3. Start the database server.

4. Ensure that the Distributed Transaction Coordinator has been started. The status can be verified in the list of services in the Computer Management console.

5. Start the Microsoft SQL Server Management Studio and connect to the local database engine as the system administrator, sa.

6. Select `File > Open > File` and select `xa_install.sql` from the subdirectory of the downloaded and extracted JDBC driver.

7. Execute the script by selecting `Query > Execute`. **Note:** Any warnings that appear in the messages section of the application window that say that stored procedures cannot be found can be safely ignored.

8. For Microsoft SQL Server JDBC drivers: If you are running Windows XP SP2, Windows XP 64-Bit Edition, or Windows Server 2003, refer to the Registry Entries Are Required for XA Transaction Support document for information on a new security constraint and how to set SQL Server on Windows XP SP2, Windows XP 64-Bit Edition, or Windows Server 2003. Create an additional value in the Windows registry for WebSphere Portal by following these steps:

   A. Open the Windows Registry Editor (regedit) and navigate to the element `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSDTC\XADLL`

   B. From the menu bar, select `Edit > New > String Value` to create a new parameter named `sqljdbc_xa.dll` in that element.

   C. Change the value of the new parameter to the location of the `sqljdbc_xa.dll` file copied in Step 2 above, for example: `C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqljdbc_xa.dll`

---

### Enable XA Transactions in Windows Component Services

1. From your Windows desktop, follow these steps:

2. Click `Start > Settings > Administrative Tools > Component Services`. 

---

330
3. Expand the tree view to locate the computer where you want to turn on support for XA transactions (for example, My Computer).
4. Display the context menu for the computer name and click **Properties**.
5. Click **Options** and tune the Transaction Timeout that suits your environment. (The recommended minimum is 180 seconds).
6. Click **MSDTC** and click **Security Configuration**.
7. Under Security Settings, select **XA Transactions** to enable this support.
8. Click OK to save your changes.

**Note:** The installation documentation for JDBC XA connectivity refers to two known problems, see the Microsoft support site for more information:
- KB318818: Performance slows down when you use XA Transactions with SQL Server.

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server

**Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - **Installing SQL Server**

  The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

  **Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
  - WebSphere Portal 6.1.5 wkplc.properties file reference
  - WebSphere Portal 6.1.5 wkplc_comp.properties file reference
  - WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**

- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:

  - `release.DbName` = `release`
  - `jcr.DbName` = `jcrdb`
  - `feedback.DbName` = `fdbkdb`
  - `likeminds.DbName` = `lmdb`
  - `community.DbName` = `commdb`
  - `customization.DbName` = `custdb`

  - If you are using a remote database, enter the values for the remote server.

  - Use a forward slash (/) instead of a backslash (\).

  - There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.

  - The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.

  - Depending on which database domain has to be configured, replace `dbdomain` with:

    - `release`
    - `customization`
    - `community`
    - `jcr`
    - `feedback`
    - `likeminds`

  - The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:

    - `dbdomainDbType`
    - `dbdomain.DbName`
- dbdomain.DbUrl
- dbdomain.DbSchema

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:

   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.

   A. For dbdomainDbType, type sqlserver2005.

   B. For dbdomainDbName, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the dbdomain.DbUrl property.

   C. For dbdomainDbSchema, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.

   D. For dbdomainDataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:

      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback

   E. For dbdomainDbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:** The database element of this value should match the value of DbName.

   F. For dbdomainDbUser, type the user ID for the database administrator.

   G. For dbdomainDbPassword, type the password for the database administrator.

   H. For dbdomainDBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.

   I. For dbdomainDBA.DbPassword, type the database administrator password for privileged access operations during creation of the database.

   J. For dbdomainDbHome, type the root location for the database. **Note:**

      - This value is the location to store the database files locally.
      - This path must use two backslashes (\) instead of a forward slash (/).

   K. For dbdomainAdminUrl, type the sqlserver URL without a database attached. This value is used to connect to the server for database administration operations.

   L. For dbdomainDbHostName, type the hostname of the database.
3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For `source.domain.DbType`, type of the database you are currently configured to use. The value for `source.domain.DbType` is Derby by default.

B. For `source.domain.DbName`, type the name of the database domain you are currently using.

C. For `source.domain.DbSchema`, type current schema identifier for objects within the database for this domain.

D. For `source.domain.DataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For `source.domain.DbUrl`, type the url currently used to access your database.

F. For `source.domain.DbUser`, type the name of the user accessing this database.

G. For `source.domain.DbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

A. For `sqlserver2005.DbDriver`, type the name of the class that SqlProcessor uses to import SQL files.

B. For `sqlserver2005.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

C. For `sqlserver2005.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

D. For `sqlserver2005.DbConnectionPoolDataSource`, type the name of the implementation class of the connection pool data source.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server  
**Previous topic:** Installing SQL Server  
**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

Before you begin, ensure that the following prerequisites are met:
- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

**Prerequisites**
- Installing SQL Server
- Modifying database properties

1. Change to the directory `wp_profile_root\ConfigEngine`
2. To create the databases, type the following command:
   ```bash
   ConfigEngine.bat create-database -DWasPassword=password
   ```
3. To create the database users, type the following command:
   ```bash
   ConfigEngine.bat setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:
   `<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql`.

**Creating users manually**

Use this alternative method for creating users if you have problems running the setup-database task that is documented for setting up a remote SQL Server 2005 database on Windows for a stand-alone production server.

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server

**Previous topic:** Modifying database properties

**Next topic:** AIX stand-alone server: Assigning custom filegroups on SQL Server 2005
Creating users manually

Use this alternative method for creating users if you have problems running the setup-database task that is documented for setting up a remote SQL Server 2005 database on Windows for a stand-alone production server.

Before you begin:

- You should have completed installing SQL Server 2005 and creating databases.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

Create the recommended database users with the SQL Server Management Studio. At least one user is required for each SQL Server 2005 instance.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Connect to your SQL Server 2005 instance.
2. Expand the tree view beneath the SQL Server instance.
3. Expand Security and right-click on Logins.
4. In the opening context menu, select New Login....
5. Enter the database user names.
7. Set a password for the selected user.
8. In the Database Access window, select the database the user must connect to at runtime. The following mappings are recommended: Table 1. List of mappings by database

<table>
<thead>
<tr>
<th>Database</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELEASE</td>
<td>RELEASEUSR</td>
</tr>
<tr>
<td>COMMUNITY</td>
<td>COMMUNITYUSR</td>
</tr>
<tr>
<td>CUSTOMIZATION</td>
<td>CUSTOMIZATIONUSR</td>
</tr>
<tr>
<td>WORKFLOW</td>
<td>WORKFLOWUSR</td>
</tr>
<tr>
<td>WMM</td>
<td>WMMDBUSR</td>
</tr>
<tr>
<td>JCRDB</td>
<td>JCR</td>
</tr>
<tr>
<td>FDBKDB</td>
<td>FEEDBACK</td>
</tr>
<tr>
<td>LMDB</td>
<td>LIKEMINDS</td>
</tr>
</tbody>
</table>

9. Click OK to save the user changes.
10. Add the role SqlJDBCXAUser to all the database users that connect to the database using XA connections. This role is available for the database "master" only. To grant users this role:
   A. Open the Microsoft SQL Server Management Studio and connect to the database instance.
   B. Expand Security > Logins underneath the database instance name.
   C. Click the user name to open the Login Properties window.
   D. Select User Mapping.
E. Select database master in the database list.
F. Select **SqlJDBCXAUser**.
G. Click **OK**.

**Parent topic:** Setting up databases
AIX stand-alone server: Assigning custom filegroups on SQL Server 2005

The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

Before you begin:

- The custom filegroups must exist prior to the execution of the database transfer.
- To see which table spaces can be customized in each domain, refer to the \wp_profile_root\PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating filegroups refer to the documentation for the database.

**Prerequisites**

- Installing SQL Server
- Modifying database properties
- Setting up databases

If custom filegroups are assigned, each must be assigned explicitly. The default filegroups can be used to contain database objects; however the name of the default file group must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom filegroups:

1. Determine the names of your custom filegroups.
2. Open the mapping file \wp_profile_root\PortalServer/config/tablespaces that specifies the table space and index space for each property pairs for each database table:
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.indexspace.indexspace
   
   For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds
3. Assign a filesystem to each entry in the mapping file. The filegroup name must be prepended by the keyword ON and a space. For example: community.COMP_INST.tablespace=ON COMM8KSPACE
   
   Repeat this step for each domain that you are transferring.
4. Save and close dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
   - UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server  
**Previous topic:** Setting up databases  
**Next topic:** Configuring WebSphere Portal to use SQL Server 2005
Configuring WebSphere Portal to use SQL Server 2005

View information on manually transferring data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- Prerequisites
  - Installing SQL Server
  - Modifying database properties
  - Setting up databases

Tips:

- If you are transferring from Oracle, the **open_cursors** setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

Steps for transferring data to another supported database

1. Open a command prompt and change to the directory `wp_profile_root\ConfigEngine`.
2. Enter the following commands to validate configuration properties.

   **ConfigEngine.bat validate-database-driver**
   
   ```
   DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ```

   **ConfigEngine.bat validate-database-connection**
   
   ```
   DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ```

3. From the same command prompt as the previous steps, change to the directory `wp_profile_root\bin`.

4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td>stopServer.bat server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td>stopServer.bat WebSphere_Portal -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

5. Transfer the database: **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory `wp_profile_root\ConfigEngine`.
B. Enter the following command:
   `ConfigEngine.bat database-transfer -DWasPassword=password`

   **Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:
   `ConfigEngine.bat database-transfer -DTransferDomainList=jcr -DWasPassword=password`

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

   If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. Change to the directory `wp_profile_root\bin`.

7. Enter the following command to start the WebSphere Portal server:
   `startServer.bat WebSphere_Portal`

8. Update the SQL Server 2005 statistics for Portal, and JCR databases by opening SQL Server Management Studio, selecting **New Query**, and running the following query:
   `use db_name
eexec sp_updatestats @resample='resample';`

Parent topic: Setting up a remote SQL Server database on Windows for a stand-alone production server

Previous topic: AIX stand-alone server: Assigning custom filegroups on SQL Server 2005
Verifying databases

After you configure IBM® WebSphere® Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the `SystemOut.log` and `SystemErr.log` files.

- **Prerequisites**
  - [Technotes for database connectivity issues](#)

You can verify the database connection using IBM WebSphere Application Server or by opening WebSphere Portal in a browser.

- To verify that the WebSphere Portal application server is running by using WebSphere Application Server, complete these steps:
  1. Open the WebSphere Application Server administrative console by entering the following address in a browser:
     ```
     http://hostname.example.com:10027/ibm/console
     ```
     where `hostname.example.com` is the fully qualified host name of the machine where WebSphere Portal is running and 10027 is the default transport port that is created by WebSphere Application Server.
  2. Log into the administrative console.
  3. Depending on your version of WebSphere Application Server, click the appropriate option:
     - For **WebSphere Application Server Version 6.1**: Click `Resources & JDBC Providers`.
     - For **WebSphere Application Server Version 7.0**: Click `Resources > JDBC > JDBC Providers`
  4. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
  5. Select the name of the data source that is defined in `wkplc_comp.properties`. The default data source is `wpdbDS`.
  6. Select the name of the JDBC provider that is specified in `wkplc_dbtype.properties`. The default JDBC provider is `wpdbJDBC_dbtype`, where `dbtype` is replaced by the value that matches your environment.
  7. Click **Test Connection** to verify the database connection. If configuration parameters have been changed, you might need to restart WebSphere Application Server for the test to complete.

- To verify that the WebSphere Portal application server is running by opening WebSphere Portal in a browser, enter the following URL in a supported browser:
  ```
  http://hostname.example.com:10040/wps/portal
  ```
  where `hostname.example.com` is the fully qualified host name of the machine where WebSphere Portal is running and 10040 is the default transport port that is created by WebSphere Application Server.

Parent topic: Configuring databases
Preparing a remote Web server when portal is installed on AIX

Install and configure the Web server plug-in, provided by IBM® WebSphere® Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

- **Prerequisites**
  - Preparing your AIX operating system
  - Installing WebSphere Portal on AIX
  - Configuring databases

Perform the following steps to install and configure your Web server:

1. Install and configure the Web server; refer to the Web server documentation for information.
2. If using IBM Lotus® Domino®, edit the `NOTES.INI` file on the Web server. Set the `HTTPEnableConnectorHeaders` parameter to 1.
3. If using IBM HTTP Server or Apache Server, edit the `httpd.conf` file on the Web server. Set the `AllowEncodedSlashes` directive to `On`; the directive should be added at the root level as a global directive.
4. Stop the Web server.
5. Install and configure the Web server plug-in on the system where the Web server is located using the Plug-ins installation wizard provided with WebSphere Application Server. Refer to Selecting a Web server topology diagram and roadmap for information. **Note:** The WebSphere Application Server Web server instructions are for the default profile. If not using the default profile, refer to Configuring a Web server for a non-default profile.

   **If using WebDAV:** After successfully installing the Web server plug-in, locate and open your `plugin-cfg.xml` file and set `AcceptAllContent` to `true`.

   **Important:** Depending on how you use the Web server, you may need to adjust the `ServerIOTimeout` value, which defines how long the plug-in should wait for a response from the application. The recommended minimum value is 60 but you may need to adjust this value higher if you are retrieving data from a database. To update this value, locate and open your `plugin-cfg.xml` file and set `ServerIOTimeout` to a value that is appropriate for your business needs. For additional information, see Common questions about the Web server plug-in.

6. If you are using Sun Web Server Version 7 update 8, read Technote 1448262 and perform the steps to resolve the HTTP 408/409 error.
7. Start the Web server.

---

**Parent topic:** Setting up a stand-alone server on AIX

**Previous topic:** Configuring databases

**Next topic:** Configuring WebSphere Portal to use a user registry on AIX

---

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>See the appropriate HTTP Server documentation</td>
<td>IBM HTTP Server</td>
</tr>
<tr>
<td>See the appropriate Apache Server documentation</td>
<td>AllowEncodedSlashes directives</td>
</tr>
</tbody>
</table>
Configuring WebSphere Portal to use a user registry on AIX

Configure user registry security on IBM® WebSphere® Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

Prior to configuring security, you should use the IBM WebSphere Application Server backupConfig task to create and store a backup of the IBM WebSphere Portal configuration; see backupConfig command for information.

- **Prerequisites**
  - Preparing your AIX operating system
  - Installing WebSphere Portal on AIX
  - Configuring databases
  - Preparing a remote Web server when portal is installed on AIX

Perform the following tasks to configure WebSphere Portal to use a user registry:

1. **Preparing user registries on AIX**
   Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

2. **Choosing your user registry model on AIX**
   Choose between securing IBM WebSphere Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

3. **Adapting the attribute configuration**
   After installing IBM WebSphere Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

4. **Stand-alone server: Configuring WebSphere Portal to use dynamic groups on AIX**
   By default, WebSphere Portal is enabled for static groups. However, the Virtual Member Manager (VMM) allows users to be members of either static or dynamic groups. Static groups are those where a persistent binding exists between a group and its members. Dynamic groups are those where a search query is defined to retrieve the members of a group.
   If you have your LDAP server configured to use dynamic groups, complete the steps in this task for WebSphere Portal to use dynamic group queries when you setup your LDAP server.

5. **Stand-alone server: Enabling referrals for your LDAP user registry on AIX**
   Referrals redirect object requests from one LDAP server to another when objects do not exist or cannot be located in a particular directory tree. You should enable referrals if your environment has more than one user registry existing on multiple servers or domains.

**Parent topic:** Setting up a stand-alone server on AIX

**Previous topic:** Preparing a remote Web server when portal is installed on AIX

**Next topic:** Tune your servers

**Related tasks**
Preparing user registries on AIX

Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

Choose the appropriate LDAP server to install and setup:

- **Preparing a Domino Directory server**
  If you plan to use a Domino Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a SecureWay Security Server**
  If you plan to use a SecureWay Security Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Novell eDirectory**
  If you plan to use a Novell eDirectory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Sun Java System Directory Server**
  If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Tivoli Directory Server**
  If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

**Parent topic:** Configuring WebSphere Portal to use a user registry on AIX  
**Next topic:** Choosing your user registry model on AIX
Preparing a Domino Directory server

If you plan to use a Domino Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Domino Directory:

1. Perform the following steps to install the Domino Directory:
   A. Go to Lotus Domino documentation.
   B. Select the appropriate version tab for your product.
   C. Click the Download/View online link for the Lotus Domino for multiple platforms Information Center.
   D. Click Domino Administrator Help > Installation > Installing and setting up Domino servers > Server installation > Installing Domino on Unix and perform this task.
   E. Click Domino Administrator Help > Installation > Installing and setting up Domino servers > The Domino server setup program and perform this task.

2. Perform the following steps as a guide to create the WebSphere Portal administrative user:
   A. Navigate to the People view of the Domino Directory and then click Add Person.
   B. Enter the following values in the New Person form to create the wpsbind user:
      - Last Name
        - wpsbind
      - User Name
        - wpsbind/DominoDomain, where DominoDomain is your Lotus Domino Internet domain
        - wpsbind Note: Make sure you enter two values in the User Name field, where the first value includes the Lotus Domino domain.
      - Short name/UserID
        - wpsbind
      - Internet password
        - wpsbind
   C. Click Save and Close to save the new person record for wpsbind and return to the People view.
   D. Click Add Person and enter the following values in the New Person form to create the wpsadmin user:
      - Last Name
        - wpsadmin, where wpsadmin in the user ID for the WebSphere Portal Administrator
      - User Name
        - wpsadmin/DominoDomain, where DominoDomain is your Lotus Domino Internet domain
        - wpsadmin Note: Make sure you enter two values in the User Name field, where the first value includes the Lotus Domino domain.
      - Short name/UserID
        - wpsadmin
      - Internet password
        - wpsadmin
   E. Click Save and Close to save the new person record for wpsadmin and return to the People view.
   F. Navigate to the Groups view and click Add Group.
   G. Enter the following values in the New Group form on the Basic tab:
      - Group name
- **wpsadmins**
  
  **Note:** If you plan to configure WebSphere Portal for multiple user registries and your Lotus Domino LDAP will share a realm with another user registry, you must use the hierarchical naming convention for the group names, for example: wpsadmins/DominoDomain, to avoid unexpected results during WebSphere Portal runtime.

- **Group type**
  - Multi-purpose

- **Members**
  - wpsbind/DominoDomain
  - wpsadmin/DominoDomain

  **Note:** You can add additional administrator users if required.

H. Click **Save and Close** to save the wpsadmins group with the wpsbind and wpsadmin users as members.

3. Perform the following steps to update the access control list for the Domino Directory:
   
   A. Open the names.nsf file in the Lotus Domino Administrator or Lotus Notes client.
   
   B. Click **File > Application > Access Control** from the main menu to open the access control list for the file.
   
   C. In the **Access Control List > Basics** panel, ensure that the wpsadmins group has either Author or Editor access.
   
   D. Add the following Role Types to the wpsadmins group:
     - GroupCreator
     - GroupModifier
     - UserCreator
     - UserModifier
   
   E. Click **OK**.

**Parent topic:** Preparing user registries on AIX
Preparing a SecureWay Security Server

If you plan to use a SecureWay Security Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare SecureWay Security Server:

1. Install SecureWay Security Server; refer to the IBM SecureWay™ Security Server for z/OS® and OS/390® for information.

2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the **Server Administration** folder, located in the left-hand navigation of the directory server console.
      2. Click the **Manage Server Properties** folder under the Server Administration folder and then select **Suffixes** on the main page.
      3. Type the **Base DN** name for the suffix; for example: `dc=yourcompany,dc=com`.
      4. Click **Add**.
      5. Click **OK** to save your changes.
      6. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the `PortalUsers.ldif` file as a working example and adapted appropriately to work with your LDAP server.
      - Use the `ContentUsers.ldif` file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every `dc=yourco,dc=com` with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the `objectclasses` to `accessGroup`. If using Tivoli Access Manager Version 6, set the `objectclasses` to `groupOfNames`.
   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.

**Parent topic:** Preparing user registries on AIX
Preparing a Novell eDirectory

If you plan to use a Novell eDirectory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Novell eDirectory:
1. Install Novell eDirectory; refer to eDirectory for information.
2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the **Server Administration** folder, located in the left-hand navigation of the directory server console.
      2. Click the **Manage Server Properties** folder under the Server Administration folder and then select **Suffixes** on the main page.
      3. Type the **Base DN** name for the suffix; for example: `dc=yourcompany,dc=com`.
      4. Click **Add**.
      5. Click **OK** to save your changes.
      6. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the **PortalUsers.ldif** file as a working example and adapted appropriately to work with your LDAP server.
      - Use the **ContentUsers.ldif** file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every `dc=yourco,dc=com` with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the **objectclasses** to **accessGroup**. If using Tivoli Access Manager Version 6, set the **objectclasses** to **groupOfNames**.
   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on AIX
Preparing a Sun Java System Directory Server

If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Sun Java System Directory Server:

1. Install Sun Java System Directory Server; refer to Oracle software for information.

2. Perform the following steps to create the WebSphere Portal administrative user:

   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the **Server Administration** folder, located in the left-hand navigation of the directory server console.
      2. Click the **Manage Server Properties** folder under the Server Administration folder and then select **Suffixes** on the main page.
      3. Type the **Base DN** name for the suffix; for example: `dc=yourcompany,dc=com`.
      4. Click **Add**.
      5. Click **OK** to save your changes.
      6. Stop and restart the LDAP server.

   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the **PortalUsers.ldif** file as a working example and adapted appropriately to work with your LDAP server.
      - Use the **ContentUsers.ldif** file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.

   C. Replace every `dc=yourco,dc=com` with your suffix.

   D. Replace any prefixes and suffixes that are unique to your LDAP server.

   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.

   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the **objectclasses** to **accessGroup**. If using Tivoli Access Manager Version 6, set the **objectclasses** to **groupOfNames**.

   G. Save your changes.

   H. Follow the instructions provided with your directory server to import the LDIF file.

   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on AIX
Preparing a Tivoli Directory Server

If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Tivoli Directory Server:

   
   **Note:** Due to a restriction in Tivoli Directory Server, users or groups must not contain a Turkish uppercase dotted I or lowercase dotted i in the DN as this will prevent correct retrieval of that user or group.

2. Perform the following steps to create the WebSphere Portal administrative user:
   
   A. Optional: Perform the following steps to create a new directory suffix:
      
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: `dc=yourcompany,dc=com`.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.

   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.

   C. Replace every `dc=yourco,dc=com` with your suffix.
   
   D. Replace any prefixes and suffixes that are unique to your LDAP server.

   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.

   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupName.
   
   G. Save your changes.

   H. Follow the instructions provided with your directory server to import the LDIF file.

   I. Stop and restart the LDAP server.

**Parent topic:** Preparing user registries on AIX
Choosing your user registry model on AIX

Choose between securing IBM® WebSphere® Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

- Prerequisites
  - Preparing user registries on AIX

Choose one of the following user registry models:

- Configuring a stand-alone LDAP user registry on AIX
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- Configuring the default federated repository on AIX
  Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

Parent topic: Configuring WebSphere Portal to use a user registry on AIX
Previous topic: Preparing user registries on AIX
Next topic: Adapting the attribute configuration
Configuring a stand-alone LDAP user registry on AIX

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Configuring a stand-alone LDAP user registry on AIX**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry to store all user account information for authorization.

- **Configuring a stand-alone LDAP user registry over SSL on AIX**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

Parent topic: Choosing your user registry model on AIX
Configuring a stand-alone LDAP user registry on AIX

Configure IBM® WebSphere® Portal to use a stand-alone LDAP user registry to store all user account information for authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps. If you need to rerun the wp-modify-ldap-security task to change the LDAP repositories or because the task failed, you must choose a new name for the realm using the standalone.ldap.realm parameter or you can set ignoreDuplicateIDs=true in the wkplc.properties file, before rerunning the task.

Perform the following steps to configure a stand-alone LDAP user registry:

**Note:** Use the wp_security_xxx.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_security_xxx.properties helper file.

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

2. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM Stand-alone LDAP configuration heading:

   - standalone.ldap.id
   - standalone.ldap.host
   - standalone.ldap.port
   - standalone.ldap.blindDN
   - standalone.ldap.bindPassword
   - standalone.ldap.ldapServerType
   - standalone.ldap.userIdMap
   - standalone.ldap.groupIdMap
   - standalone.ldap.groupMemberIdMap
   - standalone.ldap.userFilter
   - standalone.ldap.groupFilter
   - standalone.ldap.serverId
   - standalone.ldap.serverPassword
   - standalone.ldap.realm
   - standalone.ldap.primaryAdminId
   - standalone.ldap.primaryAdminPassword
   - standalone.ldap.primaryPortalAdminId
   - standalone.ldap.primaryPortalAdminPassword
   - standalone.ldap.primaryPortalAdminGroup
   - standalone.ldap.baseDN
3. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

- `standalone.idap.et.group.objectClasses`
- `standalone.idap.et.group.objectClassesForCreate`
- `standalone.idap.et.group.searchBases`
- `standalone.idap.et.personaccount.objectClasses`
- `standalone.idap.et.personaccount.objectClassesForCreate`
- `standalone.idap.et.personaccount.searchBases`

4. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attributes heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

- `standalone.idap.gm.groupMemberName`
- `standalone.idap.gm.objectClass`
- `standalone.idap.gm.scope`
- `standalone.idap.gm.dummyMember`

5. Required: Enter a value for the following required relative distinguished name (RDN®) parameters in the `wkplc.properties` file under the Default parent, RDN attribute heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

- `standalone.idap.personAccountParent`
- `standalone.idap.personAccountParentRdnProperties`
- `standalone.idap.groupParent`
- `standalone.idap.groupRdnProperties`

6. Save your changes to the `wkplc.properties` file.

7. Optional: Enter the following Web content authors parameters in the `wkplc_comp.properties` file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. **Note:** See the `wkplc_comp.properties` file for specific information about the required parameters and for advanced parameters.

- `WcmContentAuthorsGroupId`
- `WcmContentAuthorsGroupCN`

8. Run the `./ConfigEngine.sh validate-standalone-ldap -DWasPassword=password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, **WasPassword** is the value entered during installation and not a value found in your LDAP user registry. **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press y then **Enter**.

9. Run the `./ConfigEngine.sh wp-modify-ldap-security -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to set the stand-alone LDAP user registry.

10. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

11. Run the `./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

12. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the
LDAP directory.

Note: This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the configure-express task.

A. Edit the `wp_profile_root` file.

   /PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties file.

B. Add the following lines to the file:

   `uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN`  
   `cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN`

   Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `./ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory. Note: Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the wkplc.properties file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the wkplc.properties file. If the value for <code>federated.realm</code> is empty, use defaultWIMFileBasedRealm as the default value.</td>
</tr>
</tbody>
</table>

13. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

Parent topic: Configuring a stand-alone LDAP user registry on AIX

Related tasks
   Adapting the attribute configuration
   Using the member fixer tool with IBM Lotus Web Content Management
   Starting and stopping servers, deployment managers, and node agents

|   |   |   |   |
Configuring a stand-alone LDAP user registry over SSL on AIX

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to configure a standalone LDAP user registry over SSL: Note: Use the wp_security_xxx.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_security_xxx.properties helper file.

1. To specify the LDAP server’s SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:

   - **Server trust store**

   A. Add the certificate to the trust store:

      1. Log in to the WebSphere Application Server Administrative Console.
      2. Navigate to Security > SSL certificate and key management > SSL configurations.
      3. Click the appropriate SSL configuration from the list. For example,
         - Stand-alone environments: NodeDefaultSSLSettings
         - Clustered environments: CellDefaultSSLSettings

         **Clustered environments:** Ensure the setting for SSL configuration for outbound connection matches your SSL settings.

      4. Click Key stores and certificates.
      5. Click the appropriate trust store from the list; for example, NodeDefaultTrustStore.
      6. Click Signer certificates, click Add, and then enter the following information:
         - Type the Alias the key store uses for the signer certificate.
         - Type the File name where the signer certificate is located.
      7. Click OK and then click Save to save the changes to the master configuration.

   B. Retrieve the certificate from the port:

      1. Log in to the WebSphere Application Server Administrative Console.
      2. Navigate to Security > SSL certificate and key management > SSL configurations.
      3. Click the appropriate SSL configuration from the list. For example,
- Clustered environments: **CellDefaultSSLSettings**

**Clustered environments:** Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.

4. Click **Key stores and certificates**.

5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.

6. Click **Signer certificates**, click **Retrieve from port**, and then enter the following information:
   - Type the **Host** name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the **Port** used when attempting to retrieve the signer certificate.
   - Type the **Alias** the key store uses for the signer certificate.

7. Click **Retrieve signer information** to retrieve the certificate from the port.

8. Click **OK** and then click **Save** to save the changes to the master configuration.

**Client trust store**

- **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

A. See Secure installation for client signer retrieval.

B. Run the retrieveSigners task from the **wp_profile_root/bin** directory; see retrieveSigners command for information. In a deployed environment, you will need to run the retrieveSigners task, for any federated node, against the Deployment Manager. **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

C. Update the trust store properties file:

1. Use a text editor to open the **ssl.client.props** file, located in the **wp_profile_root/properties** directory.

2. Change the **com.ibm.ssl.trustStore** parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter
   ```
   com.ibm.ssl.trustStore=${CONFIG_ROOT}\cells\wpsbvt\nodes\wpsbvt\trust.p12
   ```
   to use the default trust store.

3. Save your changes.

2. Use a text editor to open the **wkplc.properties** file, located in the **wp_profile_root/ConfigEngine/properties** directory.

3. Required: Enter a value for the following required parameters in the **wkplc.properties** file under the VMM Standalone LDAP configuration heading. **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - **standalone.idap.id**
   - **standalone.idap.host**
   - **standalone.idap.port**
   - **standalone.idap.blindDN**
   - **standalone.idap.bindPassword**
   - **standalone.idap.serverType**
   - **standalone.idap.userldMap**
   - **standalone.idap.groupldMap**
   - **standalone.idap.groupMemberldMap**
   - **standalone.idap.userFilter**
   - **standalone.idap.groupFilter**
   - **standalone.idap.serverId**
   - **standalone.idap.serverPassword**
   - **standalone.idap.realm**
   - **standalone.idap.primaryAdminId**
   - **standalone.idap.primaryAdminPassword**
   - **standalone.idap.primaryPortalAdminId**
standalone.ldap.primaryPortalAdminPassword
standalone.ldap.primaryPortalAdminGroup
standalone.ldap.baseDN

4. Required: Enter a value for the following required entity types parameters in the wkplc.properties file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.et.group.objectClasses
   - standalone.ldap.et.group.objectClassesForCreate
   - standalone.ldap.et.group.searchBases
   - standalone.ldap.et.personaccount.objectClasses
   - standalone.ldap.et.personaccount.objectClassesForCreate
   - standalone.ldap.et.personaccount.searchBases

5. Required: Enter a value for the following required group member parameters in the wkplc.properties file under the Group member attributes heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.gm.groupMemberName
   - standalone.ldap.gm.objectClass
   - standalone.ldap.gm.scope
   - standalone.ldap.gm.dummyMember

6. Required: Enter a value for the following required relative distinguished name (RDN®) parameters in the wkplc.properties file under the Default parent, RDN attribute heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.personAccountParent
   - standalone.ldap.groupParent
   - standalone.ldap.personAccountRdnProperties
   - standalone.ldap.groupRdnProperties

7. Enter a value for the following parameters to enable Secure Socket Layers (SSL): **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - Required parameters:
     - standalone.ldap.sslEnabled
     - standalone.ldap.sslConfiguration
   - Optional parameters:
     - standalone.ldap.certificateMapMode
     - standalone.ldap.certificateFilter

8. Save your changes to the wkplc.properties file.

9. Run the `./ConfigEngine.sh validate-standalone-ldap -DWasPassword=password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, **WasPassword** is the value entered during installation and not a value found in your LDAP user registry.

   **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press `y` then Enter.

10. Run the `./ConfigEngine.sh wp-modify-ldap-security -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to set the stand-alone LDAP user registry.

11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

12. Run the `./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured
LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

**Parent topic:** Configuring a stand-alone LDAP user registry on AIX

**Related tasks**

Adapting the attribute configuration
Starting and stopping servers, deployment managers, and node agents
Configuring the default federated repository on AIX

Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

Perform the following tasks to add user registries to the default federated repository:

- **Configuring a federated LDAP user registry on AIX**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Adding a database user registry on AIX**
  Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

- **Adding realm support on AIX**
  A realm is a group of users from one or more user registries that form a coherent group within IBM WebSphere Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

**Parent topic:** Choosing your user registry model on AIX
Configuring a federated LDAP user registry on AIX

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Adding an LDAP user registry on AIX**
  Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time. If IBM Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=groupName` and the hierarchical format is `cn=groupName,o=root`. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as `wpsadmin`, this ID cannot exist in the LDAP you are adding.

- **Adding an LDAP user registry over SSL on AIX**
  Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

**Parent topic:** Configuring the default federated repository on AIX

---

**Notes:**
- Operating systems: AIX, HP-UX, i5/OS, Linux, Solaris, Windows
Adding an LDAP user registry on AIX

Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

If IBM® Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=groupName` and the hierarchical format is `cn=groupName,o=root`. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as `wpsadmin`, this ID cannot exist in the LDAP you are adding.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

1. Use a text editor to open the `wkplc.properties` file, located in the `/wp_profile_root/ConfigEngine/properties` directory.
2. Required: Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated LDAP Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `federated.ldap.id`
   - `federated.ldap.host`
   - `federated.ldap.port`
   - `federated.ldap.bindDN`
   - `federated.ldap.bindPassword`
   - `federated.ldap.ldapServerType`
   - `federated.ldap.baseDN`
3. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `federated.ldap.et.group.objectClasses`
   - `federated.ldap.et.group.objectClassesForCreate`
   - `federated.ldap.et.group.searchBases`
   - `federated.ldap.et.personaccount.objectClasses`
   - `federated.ldap.et.personaccount.objectClassesForCreate`
   - `federated.ldap.et.personaccount.searchBases`
4. Required: Enter a value for the following required group member parameters in the wkplc.properties file under the Group member attribute heading:

   - federated.idap.gm.groupMemberName
   - federated.idap.gm.objectClass
   - federated.idap.gm.scope
   - federated.idap.gm.dummyMember

5. Save your changes to the wkplc.properties file.

6. Run the ./ConfigEngine.sh validate-federated-ldap -DWasPassword=password task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, **WasPassword** is the value entered during installation and not a value found in your LDAP user registry.

   **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now? Press **y** then **Enter**.

7. Run the ./ConfigEngine.sh wp-create-ldap -DWasPassword=password task, from the wp_profile_root directory, to add an LDAP user registry to the default federated repository. **Note:** Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: **Starting and stopping servers, deployment managers, and node agents.**

9. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:

   A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

   B. Enter a value for the following required parameters in the wkplc.properties file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - id
      - baseDN
      - nameInRepository

   C. Save your changes to the wkplc.properties file.

   D. Run the ./ConfigEngine.sh wp-create-base-entry -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to create a base entry in a repository.

   E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Run the ./ConfigEngine.sh wp-query-repository -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to list the names and types of configured repositories.

11. Run the ./ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

12. Perform the following steps to update the user registry where new users and groups are stored:

    **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

    **Attention:** During installation, the default file repository creates a default value in the personAccountRdnProperties and groupRdnProperties parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.
A. Use a text editor to open the wkplc.properties file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. Enter a value for the following required parameters in the wkplc.properties file under the VMM supported entity types configuration heading:
   
   **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   
   - `personAccountParent`
   - `groupParent`
   - `personAccountRdnProperties`
   - `groupRdnProperties`

   The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:
   
   - `personAccountParent=dc=yourco,dc=com`
   - `groupParent=dc=yourco,dc=com`

C. Save your changes to the wkplc.properties file.

D. Run the `./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to delete the old attributes before adding the new attributes.

E. Stop and restart all necessary servers to propagate your changes.

13. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

   A. Use a text editor to open the wkplc.properties file, located in the `wp_profile_root/ConfigEngine/properties` directory.
   
   B. Enter a value for `realmName` or leave blank to update the default realm.
   
   C. Save your changes to the wkplc.properties file.
   
   D. Run the `./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.
   
   E. Stop and restart all necessary servers to propagate your changes.

14. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

   **Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

   A. Edit the `wp_profile_root/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties` file.

   B. Add the following lines to the file:

   ```
   uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
   cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
   ```

   Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

   C. Save your changes and close the file.

   D. Run the `./ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management
If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

Optional: This step is required in a production environment. Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

**Important:**
- Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
- Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

Note: If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

A. Run the following task from the `wp_profile_root/ConfigEngine` directory: 
   
   ```
   ./ConfigEngine.sh wp-change-was-admin-user - DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword
   ```
   
   **Important:** You must provide the full distinguished name (DN) for the `newAdminId` parameter.

B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

C. Run the following task from the `wp_profile_root/ConfigEngine` directory: 
   
   ```
   ./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid
   ```
   
   **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

Additional parameter for stopped servers: This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

Optional: This step is required in a production environment. Remove the file system repository if you do not use it. The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: Deleting the repository.

**Parent topic:** Configuring a federated LDAP user registry on AIX

**Related tasks**

Adapting the attribute configuration
Using the member fixer tool with IBM Lotus Web Content Management
Deleting the repository on AIX
Starting and stopping servers, deployment managers, and node agents
Adding an LDAP user registry over SSL on AIX

Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry over SSL to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

**Note:** Use the `wp_add_federated_xxx.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_add_federated_xxx.properties` helper file.

1. To specify the LDAP server's SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:
   - **Server trust store**
     A. Add the certificate to the trust store:
        1. Log in to the WebSphere Application Server Administrative Console.
        2. Navigate to **Security > SSL certificate and key management > SSL configurations**.
        3. Click the appropriate SSL configuration from the list. For example,
           - Stand-alone environments: **NodeDefaultSSLSettings**
           - Clustered environments: **CellDefaultSSLSettings**

           **Clustered environments:** Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.
        4. Click **Key stores and certificates**.
        5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.
        6. Click **Signer certificates**, click **Add**, and then enter the following information:
           - Type the **Alias** the key store uses for the signer certificate.
           - Type the **File name** where the signer certificate is located.
        7. Click **OK** and then click **Save** to save the changes to the master configuration.
   B. Retrieve the certificate from the port:
      1. Log in to the WebSphere Application Server Administrative Console.
      2. Navigate to **Security > SSL certificate and key management > SSL configurations**.
      3. Click the appropriate SSL configuration from the list. For example,
         - Stand-alone environments: **NodeDefaultSSLSettings**
- Clustered environments: **CellDefaultSSLSettings**

**Clustered environments:** Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.

4. Click **Key stores and certificates**.

5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.

6. Click **Signer certificates**, click **Retrieve from port**, and then enter the following information:
   - Type the **Host** name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the **SSL Port** used when attempting to retrieve the signer certificate.
   - Type the **Alias** the key store uses for the signer certificate.

7. Click **Retrieve signer information** to retrieve the certificate from the port.

8. Click **OK** and then click **Save** to save the changes to the master configuration.

**Client trust store**

- **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

A. See **Secure installation for client signer retrieval**.

B. Run the **retrieveSigners** task from the **wp_profile_root/bin** directory; see **retrieveSigners command** for information. In a deployed environment, you will need to run the retrieveSigners task, for any federated node, against the Deployment Manager. **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

C. Update the trust store properties file:

1. Use a text editor to open the **ssl.client.props** file, located in the **wp_profile_root/properties** directory.

2. Change the **com.ibm.ssl.trustStore** parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter
   
   ```
   com.ibm.ssl.trustStore=${CONFIG_ROOT}\cells\wpsbvt\nodes\wpsbvt\trust.p12
   ```
   to use the default trust store.

3. Save your changes.

2. Use a text editor to open the **wkplc.properties** file, located in the **wp_profile_root/ConfigEngine/properties** directory.

3. Required: Enter a value for the following required parameters in the **wkplc.properties** file under the VMM Federated LDAP Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - **federated.ldap.id**
   - **federated.ldap.host**
   - **federated.ldap.port**
   - **federated.ldap.bindDN**
   - **federated.ldap.bindPassword**
   - **federated.ldap.IdapServerType**
   - **federated.ldap.baseDN**

4. Required: Enter a value for the following required entity types parameters in the **wkplc.properties** file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - **federated.ldap.et.group.objectClasses**
   - **federated.ldap.et.group.objectClassesForCreate**
   - **federated.ldap.et.group.searchBases**
   - **federated.ldap.et.personaccount.objectClasses**
   - **federated.ldap.et.personaccount.objectClassesForCreate**
   - **federated.ldap.et.personaccount.searchBases**
5. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attribute heading:

   - `federated.idap.gm.groupMemberName`
   - `federated.idap.gm.objectClass`
   - `federated.idap.gm.scope`
   - `federated.idap.gm.dummyMember`

   **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

6. Enter a value for the following parameters to enable Secure Socket Layers (SSL):

   **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   Required parameters:
   - `federated.idap.sslEnabled`
   - `federated.idap.sslConfiguration`

   Optional parameters:
   - `federated.idap.certificateMapMode`
   - `federated.idap.certificateFilter`

7. Save your changes to the `wkplc.properties` file.

8. Optional: Enter the following Web content authors parameters in the `wkplc_comp.properties` file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. **Note:** See the `wkplc_comp.properties` file for specific information about the required parameters and for advanced parameters.

   - `WcmContentAuthorsGroupId`
   - `WcmContentAuthorsGroupCN`

9. Run the `./ConfigEngine.sh validate-federated-ldap -DWasPassword=password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, `WasPassword` is the value entered during installation and not a value found in your LDAP user registry.

   **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press `y` then `Enter`.

10. Run the `./ConfigEngine.sh wp-create-ldap -DWasPassword=password` task, from the `wp_profile_root`/ConfigEngine directory, to add an LDAP user registry to the default federated repository. **Note:** Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

12. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - `id`
      - `baseDN`
Save your changes to the wkplc.properties file.

**D.** Run the `./ConfigEngine.sh wp-create-base-entry -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to create a base entry in a repository.

**E.** Stop and restart all necessary servers to propagate your changes.

**13. Optional:** Run the `./ConfigEngine.sh wp-query-repository -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to list the names and types of configured repositories.

**14. Run the `./ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password` task,** from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

**15. Perform the following steps to update the user registry where new users and groups are stored:**

**Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

**Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

**A.** Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

**B.** Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

**Note:** See the properties file for specific information about the required parameters and for advanced parameters.

- `personAccountParent`
- `groupParent`
- `personAccountRdnProperties`
- `groupRdnProperties`

**C.** The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:
- `personAccountParent=dc=yourco,dc=com`
- `groupParent=dc=yourco,dc=com`

**D.** Save your changes to the `wkplc.properties` file.

**E.** Run the `./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to delete the old attributes before adding the new attributes.

**F.** Stop and restart all necessary servers to propagate your changes.

**16. Optional:** Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

**A.** Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

**B.** Enter a value for `realmName` or leave blank to update the default realm.

**C.** Save your changes to the `wkplc.properties` file.

**D.** Run the `./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

**E.** Stop and restart all necessary servers to propagate your changes.

**17. Optional:** Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the
Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

Note: This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the configure-express task.

A. Edit the wp_profile_root
   /PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties file.

B. Add the following lines to the file:
   
   uid=xyzadmin, o=defaultWIMFileBasedRealm -> portal_admin_DN
   cn=contentauthors, o=defaultWIMFileBasedRealm -> content_authors_group_DN

   Replace portal_admin_DN with the distinguished name of the portal administrator and content_authors_group_DN with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the ./ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password task, located in the wp_profile_root/ConfigEngine directory. Note: Choose the appropriate value to enter for realm_name depending on the type of LDAP user registry you configured: Table 1. Value for realm_name when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for realm_name should match the value for standalone ldap realm in the wkpc.properties file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for realm_name should match the value for federated realm in the wkpc.properties file. If the value for federated realm is empty, use defaultWIMFileBasedRealm as the default value.</td>
</tr>
</tbody>
</table>

18. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

19. Optional: This step is required in a production environment. Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

   Important:
   - Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
   - Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

   Note: If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

A. Run the following task from the wp_profile_root/ConfigEngine directory: ./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword Important: You must provide the full distinguished name (DN) for the newAdminid parameter.

B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.
C. Run the following task from the `wp_profile_root/ConfigEngine` directory: `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid`.

**Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

20. Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it.

The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: Deleting the repository.

**Parent topic:** Configuring a federated LDAP user registry on AIX

**Related tasks**

- Adapting the attribute configuration
- Deleting the repository on AIX
- Starting and stopping servers, deployment managers, and node agents

**Related information**

User IDs and passwords
Adding a database user registry on AIX

Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add a database user registry to the default federated repository; you must repeat these steps for each additional database user registry that you plan to add. **Note:** Use the wp_add_DB.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_add_DB.properties helper file.

1. Set up a new database, including creating a new user with appropriate database privileges for accessing the database:

   **Instructions for setting up databases:** Refer to the appropriate documentation for the type of database you want to set up.

   **Consulting your database administrator:** The task of setting up a new database is typically performed by a database administrator. However, the following steps are provided for your reference in the event you are creating a stand-alone database for testing or demonstration purposes. Consult your database administrator before proceeding with the following steps if you plan to create a database for a production environment.

   **Table 1. Steps for creating a new database to use as a database user registry.**

<table>
<thead>
<tr>
<th>Database</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DB2</strong></td>
<td>Perform the following steps to create a DB2 database: Install DB2. Enter the following database tuning commands: db2 &quot;CREATE DB dbname using codeset UTF-8 territory us PAGESIZE 8192&quot; db2 &quot;UPDATE DB CFG FOR dbname USING applheap sz 4096&quot; db2 &quot;UPDATE DB CFG FOR dbname USING app_ctl_heap_sz 1024&quot; db2 &quot;UPDATE DB CFG FOR dbname USING stmtime_heap 32768&quot; db2 &quot;UPDATE DB CFG FOR dbname USING dbheap 2400 db2 &quot;UPDATE DB CFG FOR dbname USING locklist 1000&quot; db2 &quot;UPDATE DB CFG FOR dbname USING logfilsiz 4000&quot; db2 &quot;UPDATE DB CFG FOR dbname USING logprimary 12&quot; db2 &quot;UPDATE DB CFG FOR dbname USING logsecond 20&quot; db2 &quot;UPDATE DB CFG FOR dbname USING logbufsz 32&quot; db2 &quot;UPDATE DB CFG FOR dbname USING avg_appls 5&quot; db2 &quot;UPDATE DB CFG FOR dbname USING locktimeout 30&quot; db2 &quot;UPDATE DB CFG FOR dbname using AUTO_MAINT off&quot;</td>
</tr>
</tbody>
</table>
Perform the following steps to define the `DbDriver` and `DbLibrary` parameter values:

A. Navigate to the following directory: `wp_profile_root/ConfigEngine/properties`
B. Locate and open `wkplc_dbtype.properties` with any text editor.
C. Enter a value for the following parameters under the appropriate database type properties heading:
   - `db_type.DbDriver`
   - `db_type.DbLibrary`
D. Save your changes.

3. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
4. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated Database Properties heading: **Note**: See the properties file for specific information about the required parameters and for advanced parameters.

- `federated.db.DataSourceName`
- `federated.dbDbType`
- `federated.db.DbUrl`
- `federated.db.Id`
- `federated.db.baseDN`
- `federated.db.DbUser`
- `federated.db.DbPassword`
- `federated.db.DbName`

5. Change the value for the `com.ibm.SOAP.requestTimeout` parameter to 1000.
A. Navigate to the following directory: `wp_profile_root/properties`
B. Locate and open `soap.client.props` with any text editor.
C. Locate the `com.ibm.SOAP.requestTimeout` parameter and change the value to 1000.
D. Save and close `soap.client.props`.

6. Perform the following steps in a clustered environment:
A. Run the `./ConfigEngine.sh wp-prep-vmm-db-secured-environment -DWasPassword=pass -DDbDomain=federated.db -Ddb_type.DmgrDbLibrary=local path of the database jars on the Deployment Oracle

<table>
<thead>
<tr>
<th>Oracle</th>
<th>SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perform the following steps to create an Oracle database: Install Oracle using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16. Configure the database in Dedicated Server Mode. Enter the recommended initial buffer pool sizes or set them according to your business needs.</td>
<td></td>
</tr>
<tr>
<td><code>db_block_size = 8192</code></td>
<td>Perform the following steps to create an SQL Server database: Install SQL Server. Set <strong>Collation</strong> to case-sensitive. <strong>Note</strong>: Install SQL Server with the appropriate portal database collation so that your tempdb collation setting matches the collation you use for the property extension database. The tempdb collation is inherited from the master database, which you set when you install SQL Server.</td>
</tr>
<tr>
<td><code>db_cache_size = 300M</code></td>
<td><code>federated.db.DataSourceName</code></td>
</tr>
<tr>
<td><code>db_files = 1024</code></td>
<td><code>federated.dbDbType</code></td>
</tr>
<tr>
<td><code>log_buffer = 65536</code></td>
<td><code>federated.db.DbUrl</code></td>
</tr>
<tr>
<td><code>open_cursors = 1500</code></td>
<td><code>federated.db.Id</code></td>
</tr>
<tr>
<td><code>pga_aggregate_target = 200M</code></td>
<td><code>federated.db.baseDN</code></td>
</tr>
<tr>
<td><code>pre_page_sga = true</code></td>
<td><code>federated.db.DbUser</code></td>
</tr>
<tr>
<td><code>shared_pool_size = 200M</code></td>
<td><code>federated.db.DbPassword</code></td>
</tr>
<tr>
<td><code>processes = 300</code></td>
<td><code>federated.db.DbName</code></td>
</tr>
</tbody>
</table>
Manager -DDmgrNodeName=dmgr_node_name task from the wp_profile_root/ConfigEngine directory to create the local Deployment Manager WebSphere variable used to access the database jars. **Note:** The db_type in db_type .DmgrDbLibrary should be set to the type of database you are using, for example db2. The local full path of the database jars on the Deployment Manager should be one of the following options:

- **DB2 Type 2 driver:** db2java.zip
- **DB2 Type 4 driver:** db2jcc.jar;db2jcc_license_cu.jar
- **DB2 for z/OS Type 2 driver:** db2java.zip
- **DB2 for z/OS Type 4 driver:** db2jcc.jar;db2jcc_license_cisuz.jar;db2jcc_javax.jar
- **Oracle:** ojdbc14.jar
- **SQL Server JDBC driver provided by Microsoft:** sqljdbc.jar
- **SQL Server JDBC driver provided by DataDirect:** sqlserver.jar;base.jar;util.jar

B. Run the following task. Include each node name as a comma separated list in the command:

**Running the task:** You do not have to run this task more than once. You can run this task from any node in the cluster.

1. Set the property value for federated.dbDbType in the wkplc.properties file if using a database user registry.
2. Run the ./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=password -DDbDomain=federated.db -DVmmNodeName=node_name -Db DbType.NodeDbLibrary=local full path of the database jars task from the wp_profile_root/ConfigEngine directory on each node to create the variable used to access the VMM database jars. **Note:** VmmNodeName is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The db_type in db_type.NodeDbLibrary should be set to the type of database you are using, for example db2.

C. Stop and restart all necessary servers to propagate your changes.

7. Run the ./ConfigEngine.sh wp-create-db -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add a database user registry to the default federated repository. **Note:** Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

9. Perform the following steps to update the user registry where new users and groups are stored:

**Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

**Attention:** During installation, the default file repository creates a default value in the personAccountRdnProperties and groupRdnProperties parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

B. Enter a value for the following required parameters in the wkplc.properties file under the VMM supported entity types configuration heading:

**Note:** See the properties file for specific information about the required parameters and for advanced parameters.
- **personAccountParent**
- **groupParent**
- **personAccountRdnProperties**
- **groupRdnProperties**

The parameters **groupParent** and **personAccountParent** must be set to the same value. For example:
- personAccountParent=dc=yourco,dc=com
- groupParent=dc=yourco,dc=com

C. Save your changes to the wkplc.properties file.

D. Run the ./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to delete the old attributes before adding the new attributes.

E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Run the ./ConfigEngine.sh wp-query-repository -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to list the names and types of configured repositories.

**Parent topic:** Configuring the default federated repository on AIX

**Parent topic:** Updating your user registry on AIX
Adding realm support on AIX

A realm is a group of users from one or more user registries that form a coherent group within IBM® WebSphere® Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

Before configuring realm support, you must add all LDAP user registries and/or database user registries, that you will use to create a single realm or multiple realms, to the federated repository. If you are going to create multiple realms, you must create all required base entries within your LDAP user registries and/or database user registries. All base entry names must be unique within the federated repository. In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add realm support to your user registry model:

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
2. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - realmName
   - securityUse
   - delimiter
   - addBaseEntry
3. Save your changes to the wkplc.properties file.
4. Run the ./ConfigEngine.sh wp-create-realm -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add a new realm to the Virtual Member Manager configuration. Important: To create multiple realms, ensure that your federated repository contains the required unique base entries. Stop and restart the appropriate servers for your installation environment, and then update the wkplc.properties file with the base entry information and rerun the wp-create-realm task. Repeat these steps until all realms are created.
5. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.
6. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading and then save your changes: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - realmName
   - realm.personAccountParent
   - realm.groupParent
   - realm.orgContainerParent
7. Run the ./ConfigEngine.sh wp-modify-realm-defaultparents -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to update the default parents per entity type and realm. Important: Stop and restart the appropriate servers for your installation environment before rerunning this task for any additional entity types and realms.
Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

Optional: Perform the following steps to add additional base entries to the realm configuration; for example, if you had two additional base entries (base entry 1 and base entry 2) to add to the realm you just created, you would update the wkplc.properties file with the information from base entry 1 and then run this task. Then you would update the properties file with the information for base entry 2 and then run this task:

A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

B. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading:
   - realmName
   - addBaseEntry
   Note: See the properties file for specific information about the required parameters and for advanced parameters.

C. Save your changes to the wkplc.properties file.

D. Run the .:/ConfigEngine.sh wp-add-realm-baseentry -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add additional LDAP base entries to the realm configuration.

E. Stop and restart all necessary servers to propagate your changes.

Optional: Perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID; this step is required if you change the default realm:

A. Create a new user in the Manage Users and Groups portlet to replace the current WebSphere Application Server administrative user.

B. Create a new user in the Manage Users and Groups portlet to replace the current WebSphere Portal administrative user.

C. Create a new group in the Manage Users and Groups portlet to replace the current group.

D. Run the .:/ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid task, from the wp_profile_root/ConfigEngine directory, to replace the old WebSphere Application Server administrative user ID and group ID with the new user and group. Important: You must provide the full distinguished name (DN) for the newAdminId and newAdminGroupId parameters.

   Additional parameter for stopped servers: This task verifies the user against a running server instance. If the server is stopped, add the -Dskip.ldap.validation=true parameter to the task to skip the validation.

E. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

F. Run the .:/ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid task to replace the old WebSphere Portal administrative user ID and group ID with the new user and group. Important: You must provide the full distinguished name (DN) for the newAdminId and newAdminGroupId parameters.

   Additional parameter for stopped servers: This task verifies the user against a running server instance. If the server is stopped, add the -Dskip.ldap.validation=true parameter to the task to skip the validation.

G. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

Optional: Perform the following steps to set the realm you created as the default realm: Remember: Only users defined in base entries that exist in the default realm are able to log into WebSphere Portal. If you find that a user cannot log in to WebSphere Portal, check to see if the base entry that contains the user exists in the default realm. You can run the wp-query-realm-baseentry task to see what base entries are part of the default realm. If the default realm is missing the base entry, run the wp-add-realm-baseentry task to add the base entry to the default realm.
A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. For `defaultRealmName`, type the `realmName` property value you want to use as the default realm.

C. Save your changes to the `wkplc.properties` file.

D. Run the `./ConfigEngine.sh wp-default-realm -DWasPassword=password` task, from the `wp_profile_root` directory, to set this realm as the default realm.

E. Stop and restart all necessary servers to propagate your changes.

12. Optional: Perform the following steps to query a realm for a list of its base entries:

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. For `realmName`, type the name of the realm you want to query.

C. Save your changes to the `wkplc.properties` file.

D. Run the `./ConfigEngine.sh wp-query-realm-baseentry -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to list the base entries for a specific realm.

13. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. Enter a value for `realmName` or leave blank to update the default realm.

C. Save your changes to the `wkplc.properties` file.

D. Run the `./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

E. Stop and restart all necessary servers to propagate your changes.

**Parent topic:** Configuring the default federated repository on AIX
Adapting the attribute configuration on AIX

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

**Prerequisites**

- Preparing user registries on AIX
- Choosing your user registry model on AIX

After installation, IBM WebSphere Portal has a predefined set of attributes for users and groups. Your LDAP server may have a different set of predefined user and group attributes. To ensure proper communication between WebSphere Portal and your LDAP server, you can configure additional attributes and flag existing attributes as required or unsupported on a per repository basis or for all configure repositories.

LDAP servers can only handle attributes that are explicitly defined in their schema. The LDAP server's schema is different from the WebSphere Portal schema but the two schemas should match for proper communication between WebSphere Portal and the LDAP server. The task to add the LDAP user registry does some basic attribute configurations depending on the type of LDAP server that you choose. You may, however, still need to adapt the WebSphere Portal configuration to match the LDAP schema; for example, if an attribute is defined in WebSphere Portal but not in the LDAP server, you will need to perform one of the following tasks to resolve this mismatch:

- Flag the attribute as unsupported for the LDAP server
- Introduce an attribute mapping that maps the WebSphere Portal attribute to an attribute defined in the LDAP schema

Perform the following tasks to adapt the attribute configuration to match the configured LDAP server(s) and your business needs:

1. **Querying the defined attributes on AIX**
   After installing IBM WebSphere Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

2. **Adding attributes on AIX**
   The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM WebSphere Portal and your user registry.

3. **Mapping attributes**
   After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

4. **Removing attributes**
   Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

**Parent topic:** Configuring WebSphere Portal to use a user registry on AIX

**Previous topic:** Choosing your user registry model on AIX

**Next topic:** Stand-alone server: Configuring WebSphere Portal to use dynamic groups on AIX
Related tasks
Adding an LDAP user registry on AIX
Adding an LDAP user registry over SSL on AIX
Configuring a stand-alone LDAP user registry on AIX
Configuring a stand-alone LDAP user registry over SSL on AIX
Querying the defined attributes on AIX

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

**Warning:** If you are using a database user registry or a property extension database, copy the database drivers to the `AppServer_root/lib` directory before executing this script.

Run the `./ConfigEngine.sh wp-query-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, any time during the configuration process or at runtime to query an overview of the currently defined attributes. This task creates the `availableAttributes.html` report, located in the `wp_profile_root/ConfigEngine/log` directory. The report contains one table that lists the available attributes for Users (*PersonAccount*) and one table that lists the available attributes for Groups. For each configured repository there is a column that indicates if the attribute is flagged as unsupported or if the attribute is mapped to a different LDAP attribute. **Note:** This task does not validate the existence of attributes in the LDAP schema.

**Parent topic:** Adapting the attribute configuration

**Next topic:** Adding attributes on AIX
Adding attributes on AIX

The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM® WebSphere® Portal and your user registry.

- **Prerequisites**

  - Querying the defined attributes on AIX

Perform the following steps to add new attributes to your user registry:

1. Install the required Enterprise Archive (.ear) file on WebSphere Application Server. Select the appropriate task to install the .ear file: Table 1. Steps for installing the .ear file by environment

<table>
<thead>
<tr>
<th>Environment</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stand-alone environment</strong></td>
<td>Perform the following steps to install the .ear file in a stand-alone environment: Open a command prompt. Navigate to the <code>&lt;wp_profile_root&gt;/ConfigEngine</code> directory. Run the command <code>./ConfigEngine.sh wp-la-install-ear -DWasPassword=your_password</code> task.</td>
</tr>
<tr>
<td><strong>Clustered environment</strong></td>
<td>Perform the following steps to install the .ear file in a clustered environment: Open a command prompt. Navigate to the <code>&lt;wp_profile_root&gt;/ConfigEngine</code> directory on the primary node. Run the command <code>./ConfigEngine.sh wp-la-install-ear -DWasPassword=dmgr_password -DServerName=dmgr -DNodeName=node_name</code> task. Where <code>node_name</code> is the name of the node where the deployment manager resides; you can find the <code>node_name</code> value in the WebSphere Application Server Administrative Console under System administrator &gt; Deployment Manager &gt; Runtime tab &gt; General Properties &gt; Node Name.</td>
</tr>
</tbody>
</table>

2. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

3. Use a text editor to open the `wkplc.properties` file, located in the `<wp_profile_root>/ConfigEngine/properties` directory.

4. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Property Extension Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `la.providerURL`
   - `la.propertyName`
   - `la.entityTypes`
   - `la.dataType`
   - `la.multiValued`
5. Save your changes to the `wkplc.properties` file.

6. Run the `./ConfigEngine.sh wp-add-property -DWasPassword=password` task to add the attribute to the user registry.

   **Note:** This task performs an EJB call to WebSphere Application Server, which must authenticate against WebSphere Application Server. Depending on the configuration in the `sas.client.props` file, you may receive a popup window or a command line prompt asking for user identity and password. Enter the WebSphere Application Server user ID and password.

   **Remember:** If you have multiple properties to add, repeat all steps, except for the `wp-la-install-ear` task, until all new attributes are added.

7. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents.*

**Related tasks**

- [Adapting the attribute configuration](#)
- [Querying the defined attributes on AIX](#)
- [Mapping attributes](#)

**Parent topic:** [Adapting the attribute configuration](#)

**Previous topic:** [Querying the defined attributes on AIX](#)

**Next topic:** [Mapping attributes](#)
Mapping attributes

After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

**- Prerequisites**

- Querying the defined attributes on AIX
- Adding attributes on AIX

Perform the following steps to map attributes between WebSphere Portal and your LDAP server; if you have multiple LDAP servers, you will need to perform these steps for each LDAP server:

1. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
2. Enter a value for one of the following sets of parameters in the `wkplc.properties` file to identify your LDAP server:
   
   **Note:** Make sure you use the same values you used to configure your LDAP server.

   **Table 1. Identifying your LDAP server in the wkplc.properties file.**

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>The following parameters are found under the LDAP attribute configuration heading: <strong>Note:</strong> See the properties file for specific information about the required parameters and for advanced parameters. standalone.ldap.host standalone.ldap.port standalone.ldap.sslEnabled standalone.ldap.bindDN standalone.ldap.bindPassword standalone.ldap.baseDN</td>
</tr>
<tr>
<td>Federated</td>
<td>The following parameters are found under the VMM Federated repository properties heading: <strong>Note:</strong> See the properties file for specific information about the required parameters and for advanced parameters. federated.ldap.host federated.ldap.port federated.ldap.sslEnabled federated.ldap.bindDN federated.ldap.bindPassword</td>
</tr>
</tbody>
</table>

3. Run one of the following tasks to check that all defined attributes are available in the configured LDAP user registry:

   **Table 2. Task to check that all defined attributes are available in the configured LDAP user registry.**

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td><code>./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWWasPassword=password</code> task, from the <code>wp_profile_root/ConfigEngine</code> directory.</td>
</tr>
<tr>
<td>Federated</td>
<td><code>./ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWWasPassword=password</code> task, from the <code>wp_profile_root/ConfigEngine</code> directory.</td>
</tr>
</tbody>
</table>
4. Open the ConfigTrace.log file, located in the `wp_profile_root/ConfigEngine/log` directory, to review the following output for the PersonAccount and Group entity type:

**The following attributes are defined in WebSphere Portal but not in the LDAP server**
- This list contains all attributes that are defined in WebSphere Portal but not available in the LDAP. Flag attributes that you do not plan to use in WebSphere Portal as unsupported. Map the attributes that you plan to use to the attributes that exist in the LDAP; you must also map the `uid`, `cn`, `firstName`, `sn`, `preferredLanguage`, and `ibm-primaryEmail` attributes if they are contained in the list.

**The following attributes are flagged as required in the LDAP server but not in WebSphere Portal**
- This list contains all attributes that are defined as "MUST" in the LDAP server but not as required in WebSphere Portal. You should flag these attributes as required within WebSphere Portal; see the step below about flagging an attribute as either unsupported or required.

**The following attributes have a different type in WebSphere Portal and in the LDAP server**
- This list contains all attributes that WebSphere Portal might ignore because the data type within WebSphere Portal and within the LDAP server do not match.

5. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

6. Enter a value for one of the following sets of parameters in the `wkplc.properties` file to correct any issues found in the config trace file: *Table 3. Parameters to define in the wkplc.properties file to correct any issues found in the config trace file.*

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stand-alone</strong></td>
<td>The following parameters are found under the LDAP attribute configuration heading: <strong>Note:</strong> See the properties file for specific information about the required parameters and for advanced parameters. &lt;br&gt;<code>standalone.ldap.idstandalone.ldap.attributes.nonSupportedstandalone.ldap.attributes.nonSupported.deletestandalone.ldap.attributes.mapping.ldapNamestandalone.ldap.attributes.mapping.portalNamestandalone.ldap.attributes.mapping.entityTypes</code>For example, the following values will flag certificate and members as unsupported attributes and will map <code>ibm-primaryEmail</code> to <code>mail</code> and <code>ibm-jobTitle</code> to <code>title</code> for both the PersonAccount and Group entity types. &lt;br&gt;<code>standalone.ldap.attributes.nonSupported=certificate, members</code> &lt;br&gt;<code>standalone.ldap.attributes.mapping.ldapName=ibm-primaryEmail, ibm-jobTitle</code> &lt;br&gt;<code>standalone.ldap.attributes.mapping.entityTypes=PersonAccount, Group</code></td>
</tr>
</tbody>
</table>
7. Save your changes to the `wkplc.properties` file.

8. Run one of the following tasks to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between WebSphere Portal and the LDAP user registry:

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td><code>.ConfigEngine.sh wp-update-standalone-ldap-attribute-config -DWasPassword=passwrod task, from the wp_profile_root/ConfigEngine directory</code></td>
</tr>
<tr>
<td>Federated</td>
<td><code>.ConfigEngine.sh wp-update-federated-ldap-attribute-config -DWasPassword=passwrod task, from the wp_profile_root/ConfigEngine directory</code></td>
</tr>
</tbody>
</table>

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: `Starting and stopping servers, deployment managers, and node agents`.

10. Optional: Perform the following steps to flag an attribute as either unsupported or required for the entire WebSphere Portal environment instead of just for the specified LDAP:

   A. Enter a value for the following required parameters in the `wkplc.properties` file: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
      - `user.attributes.required`
      - `user.attributes.nonsupported`

   B. Save your changes to the `wkplc.properties` file.

   C. Run the `./ConfigEngine.sh wp-update-attribute-config -DWasPassword=passwrod task, from the wp_profile_root/ConfigEngine directory`.

   D. Stop and restart all necessary servers to propagate your changes.

Parent topic: Adapting the attribute configuration

Previous topic: Adding attributes on AIX
Next topic: Removing attributes

Related tasks
Starting and stopping servers, deployment managers, and node agents
Removing attributes

Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

- **Prerequisites**
  - Querying the defined attributes on AIX
  - Adding attributes on AIX
  - Mapping attributes

Perform the following steps to remove an attribute from your database: **Important:** Do not remove attributes that have already been populated with user values because this can cause database inconsistencies. **Cluster note:** In a clustered environment, perform the following steps on the deployment manager and then resynch the nodes.

1. Perform the following steps to remove an attribute stored in a property extension database; this is an attribute that was added using the `wp-add-la-property` task.
   - A. Open the tool you use to edit your database.
   - B. Verify that your attribute name is available in the LAPROP table.
   - C. Delete the required attributes from the LAPROP table.
   - D. Open the `wimxmlextension.xml` file, located in the `wp_profile_root/config/cells/cellname/wim/model` directory.
   - E. Locate and delete the `propertySchema` definition for the attributes that you deleted from the LAPROP table; for example:

   ```xml
   <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String" multiValued="true" propertyName="attribute_name">
   <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
   </wim:propertySchema>
   ```

   - F. Save your changes to the `wimxmlextension.xml` file.
   - G. Open the `wimconfig.xml` file, located in the `wp_profile_root/config/cells/cellname/wim/config` directory.
   - H. Locate and delete the `propertiesNotSupported` definitions for the attributes that you deleted from the LAPROP table; for example:

   ```xml
   <config:propertiesNotSupported name="attribute_name"/>
   ```

   - I. Save your changes to the `wimconfig.xml` file.
   - J. Stop and restart the server1 and WebSphere_Portal servers from the `wp_profile_root/bin` directory.

2. Perform the following steps to remove an attribute that is not stored in a property extension database:
   - A. Open the `wimxmlextension.xml` file.
   - B. Locate and delete the `propertySchema` definition for the attributes you previously added:

   ```xml
   <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String" multiValued="true" propertyName="attribute_name">
   <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
   </wim:propertySchema>
   ```

   - C. Save your changes to the `wimxmlextension.xml` file.
   - D. Open the `wimconfig.xml` file.
   - E. Locate and delete the stanza that corresponds to the custom attribute you deleted from the `wimextension.xml` file; for example:

   ```xml
   <config:attributes name="attribute_name" propertyName="property_name">
   ```

   - F. Save your changes to the `wimconfig.xml` file.
F. Save your changes to the wimconfig.xml file.
G. Stop and restart the server1 and WebSphere_Portal servers.

Parent topic: Adapting the attribute configuration
Previous topic: Mapping attributes
Stand-alone server: Configuring WebSphere Portal to use dynamic groups on AIX

By default, WebSphere Portal is enabled for static groups. However, the Virtual Member Manager (VMM) allows users to be members of either static or dynamic groups. Static groups are those where a persistent binding exists between a group and its members. Dynamic groups are those where a search query is defined to retrieve the members of a group. If you have your LDAP server configured to use dynamic groups, complete the steps in this task for WebSphere Portal to use dynamic group queries when you setup your LDAP server.

Perform the required tasks to configure either a stand-alone or federated LDAP server security.

- **Prerequisites**
  - Preparing user registries on AIX
  - Choosing your user registry model on AIX
  - Adapting the attribute configuration

The steps in this task use `groupOfURLs` as the object class for dynamic groups and `memberURL` as the dynamic membership attribute. The actual values for object classes and dynamic membership attributes can vary depending on your LDAP server. For this reason, you should export an LDIF file to verify the object classes and dynamic membership attributes. Either refer to your LDAP documentation or ask your LDAP administrator for instructions on exporting an LDIF file.

**Clustered environments:** Perform the following steps on the Deployment Manager then synchronize the nodes.

To configure WebSphere Portal to use dynamic groups, do the following:

1. Choose the appropriate set of steps, depending on your LDAP server environment: *Table 1. Steps for enabling dynamic groups*

<table>
<thead>
<tr>
<th>LDAP server environment</th>
<th>Steps to perform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone LDAP server or federated LDAP server(s)</td>
<td>Navigate to the following directory: <code>wp_profile_root/cells/cell_name/wim/config</code>. Locate and open <code>wimconfig.xml</code> with any text editor. Add the following line to the <code>&lt;config:groupConfiguration&gt;</code> tag: <code>&lt;config:dynamicMemberAttributes name=&quot;memberurl&quot; objectClass=&quot;groupofurls&quot;/&gt;</code>. Save and close <code>wimconfig.xml</code>.</td>
</tr>
</tbody>
</table>

By default, WebSphere Portal is enabled for static groups. However, the Virtual Member Manager (VMM) allows users to be members of either static or dynamic groups. Static groups are those where a persistent binding exists between a group and its members. Dynamic groups are those where a search query is defined to retrieve the members of a group. If you have your LDAP server configured to use dynamic groups, complete the steps in this task for WebSphere Portal to use dynamic group queries when you setup your LDAP server.
2. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: **Starting and stopping servers, deployment managers, and node agents.**

| Federated LDAP server(s) | Log in to the administration console. Select **Security > Secure administration, applications, and infrastructure.** Under **Available realm definitions**, select **Federated repositories** and click **Configure.** Under **Related items**, click **Manage repositories.** Select the appropriate repository from the list. Under **Additional properties**, click **Group attribute definition** then click **Dynamic member attributes.** Click **New** and specify values for the **Name** and **Object class** fields as appropriate. For example, **Name**: `memberurl` **Object class**: `groupofurls` Click **OK** and save the changes to the master configuration. |

**Parent topic:** Configuring WebSphere Portal to use a user registry on AIX  
**Previous topic:** Adapting the attribute configuration  
**Next topic:** Stand-alone server: Enabling referrals for your LDAP user registry on AIX
Stand-alone server: Enabling referrals for your LDAP user registry on AIX

Referrals redirect object requests from one LDAP server to another when objects do not exist or cannot be located in a particular directory tree. You should enable referrals if your environment has more than one user registry existing on multiple servers or domains.

- **Prerequisites**
  - Preparing user registries on AIX
  - Choosing your user registry model on AIX
  - Adapting the attribute configuration
  - Stand-alone server: Configuring WebSphere Portal to use dynamic groups on AIX

To configure your portal to use LDAP referrals, do the following:

1. Use any text editor to open the wkplc.properties file in the following directory: `wp_profile_root/ConfigEngine/properties`.
2. Specify values for the following parameters:
   - `et.ldap.id=ID_of_your_LDAP_server`
   - `et.ldap.host=hostname_of_your_LDAP_server`
   - `et.ldap.referral=follow`
3. Save and close wkplc.properties.
4. Run the following task from the `wp_profile_root/ConfigEngine` directory to create an LDAP entity type:
   - UNIX: `./ConfigEngine.sh wp-update-et-ldap -DWasPassword=password`
   - Windows: `ConfigEngine.bat wp-update-et-ldap -DWasPassword=password`
   - IBM® i5/OS: `ConfigEngine.sh wp-update-et-ldap -DWasPassword=password`
5. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents.*

**Parent topic:** Configuring WebSphere Portal to use a user registry on AIX

**Previous topic:** Stand-alone server: Configuring WebSphere Portal to use dynamic groups on AIX
Tune your servers

Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

- Prerequisites
  - Preparing your AIX operating system
  - Installing WebSphere Portal on AIX
  - Configuring databases
  - Preparing a remote Web server when portal is installed on AIX
  - Configuring WebSphere Portal to use a user registry on AIX

Base Portal Tuning Scenarios

Tuning a WebSphere Portal environment involves tuning and configuring the various systems and components of the environment. The tuning guide provides general concepts and details specifics configuration instructions. Instructions are included for:

- Configuring the application server and the resources defined for that application server
- Determining the cloning strategy for expanding or extending the environment
- Tuning the database(s) and database server
- Tuning the directory server and its database
- Tuning the Web server
- Tuning the operating system and network
- Tuning the WebSphere Portal services

Parent topic: Setting up a stand-alone server on AIX
Previous topic: Configuring WebSphere Portal to use a user registry on AIX

Related information
- WebSphere Portal and Lotus Web Content Management 6.1.x Performance Tuning Guide
- IBM WebSphere Portal Performance Troubleshooting Guide
Setting up a stand-alone server on HP-UX

After you have prepared the operating system you are ready to install WebSphere Portal.

- **Prerequisites**
  - [Technotes for installation and configuration issues](#)

Select the operating system on which you are installing WebSphere Portal.

1. **Preparing your HP-UX operating system**
   View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

2. **Installing WebSphere Portal on HP-UX**
   IBM WebSphere Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

3. **Configuring databases**
   Before transferring default data to the production database server, you need to prepare the database server. Database preparation includes creating user IDs and databases. For this installation path, stand-alone production server, remote databases servers are used.

4. **Preparing a remote Web server when portal is installed on HP-UX**
   Install and configure the Web server plug-in, provided by IBM WebSphere Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

5. **Configuring WebSphere Portal to use a user registry on HP-UX**
   Configure user registry security on IBM WebSphere Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

6. **Tune your servers**
   Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

**Parent topic:** Setting up a stand-alone production server
Preparing your HP-UX operating system

View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

Configuring HP-UX kernel settings before installing

To set kernel parameters, perform the following steps:

1. Log into the host machine with superuser (root) privileges.
2. Choose one of the following options to determine the physical memory, which you must know to avoid setting certain kernel parameters above the physical capacity:
   - HP-UX 11i Version 3 and later: Use the /usr/sbin/dmesg command.
   - HP-UX 11i Version 2 and earlier: Perform the following steps:
     A. Start the HP-UX System Administration Manager (SAM) utility with the /usr/sbin/sam command.
     B. Select Performance Monitors > System Properties > Memory.
     C. View the value for Physical Memory and click OK.
     D. Exit from the SAM utility.
3. Set the maxfiles and maxfiles_lim parameters to at least 4096. (The table below recommends 8000 and 8196, respectively. You must first edit the /usr/conf/master.d/core-hpux file, to allow the SAM utility to set values greater than 2048: Note: If using HP-UX 11i Version 2, the /usr/conf/master.d/core-hpux file does not exist; instead, use the SAM utility to set the new kernel parameter values. See Reconfiguring the Kernel (HP-UX 11i Version 2) for details.
   A. Open the file /usr/conf/master.d/core-hpux in a text editor.
   B. Change the line, "*range maxfiles<=2048" to "*range maxfiles=60000"
   C. Change the line, "*range maxfiles_lim<=2048" to "*range maxfiles_lim=60000"
   D. Save and close the file. Old values might be stored in the /var/sam/boot.config file. Force the SAM utility to create a new boot.config file:
      1. Move the existing version of the /var/sam/boot.config file to another location, such as the /tmp directory.
      2. Start the SAM utility.
   3. Select Kernel Configuration > Configurable Parameters. When the Kernel Configuration window opens, a new boot.config file exists. Alternatively, rebuild the boot.config file with this command: #
      /usr/sam/lbin/getkinfo -b
4. Web Content Management only: Use the ulimit -f command to set the maximum size of files that can be created to be at least the size of the largest file you would need to upload to the content server. The command ulimit -f -1 removes any limit on file size.
5. Set new kernel parameter values:
   A. Start the SAM utility.
   B. Select Kernel Configuration > Configurable Parameters.
   C. For each of the parameters in the following table, perform this procedure:
      1. Highlight the parameter to change.
      2. Select Actions > Modify Configurable Parameter.
      3. Type the new value in the Formula/Value field.
      4. Click OK.

Note: When WebSphere Portal and DB2 are installed on the same machine, some kernel values are higher than those shown in the table below. If you are using this configuration, refer to the following site for more information:
Typical kernel settings for running WebSphere Portal appear in the following table:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbc_max_pct</td>
<td>25</td>
</tr>
<tr>
<td>maxdsize</td>
<td>805306358</td>
</tr>
<tr>
<td>maxdsize</td>
<td>20480000000 (when WebSphere Application Server Base and Network Deployment products are on the same machine)</td>
</tr>
<tr>
<td>maxfiles_lim</td>
<td>8196 (Change this one before maxfiles.)</td>
</tr>
<tr>
<td>maxfiles</td>
<td>8000</td>
</tr>
<tr>
<td>maxssiz</td>
<td>8388608</td>
</tr>
<tr>
<td>maxswapchunks (valid for HP-UX 11iv1 only)</td>
<td>8192</td>
</tr>
<tr>
<td>max_thread_proc</td>
<td>3000</td>
</tr>
<tr>
<td>maxuprc</td>
<td>512</td>
</tr>
<tr>
<td>maxusers (valid for HP-UX 11iv1 only)</td>
<td>512</td>
</tr>
<tr>
<td>msgmap</td>
<td>2048</td>
</tr>
<tr>
<td>msgmax</td>
<td>65535</td>
</tr>
<tr>
<td>msgmax</td>
<td>131070 (when WebSphere Application Server Base and Network Deployment products are on the same machine)</td>
</tr>
<tr>
<td>msgmnb</td>
<td>65535</td>
</tr>
<tr>
<td>msgmnb</td>
<td>131070 (when WebSphere Application Server Base and Network Deployment products are on the same machine)</td>
</tr>
<tr>
<td>msgmni</td>
<td>50</td>
</tr>
<tr>
<td>msgseg</td>
<td>32767</td>
</tr>
<tr>
<td>msgssz</td>
<td>96</td>
</tr>
<tr>
<td>msgtql</td>
<td>2046</td>
</tr>
<tr>
<td>nfile</td>
<td>58145</td>
</tr>
<tr>
<td>nflocks</td>
<td>3000</td>
</tr>
<tr>
<td>ninode</td>
<td>60000</td>
</tr>
<tr>
<td>nkthread</td>
<td>7219</td>
</tr>
<tr>
<td>nproc</td>
<td>4116</td>
</tr>
<tr>
<td>npty</td>
<td>2024</td>
</tr>
<tr>
<td>nstrpty</td>
<td>1024</td>
</tr>
<tr>
<td>nstrtel</td>
<td>60</td>
</tr>
<tr>
<td>sema (valid for HP-UX 11iv1 only)</td>
<td>1 (with Embedded Messaging)</td>
</tr>
<tr>
<td>semaem</td>
<td>16384 (with Embedded Messaging)</td>
</tr>
<tr>
<td>semmap (valid for HP-UX 11iv1 only)</td>
<td>514</td>
</tr>
<tr>
<td>semmni</td>
<td>2048</td>
</tr>
<tr>
<td>semmns</td>
<td>16384 (with Embedded Messaging)</td>
</tr>
<tr>
<td>semmnu</td>
<td>1024</td>
</tr>
<tr>
<td>semume</td>
<td>200</td>
</tr>
<tr>
<td>semvmx</td>
<td>32767 (with Embedded Messaging)</td>
</tr>
<tr>
<td>shmmmax</td>
<td>2147483647</td>
</tr>
<tr>
<td>shmem (valid for HP-UX 11iv1 only)</td>
<td>1 (with Embedded Messaging)</td>
</tr>
<tr>
<td>shmmni</td>
<td>1024</td>
</tr>
<tr>
<td>shmseg</td>
<td>1024</td>
</tr>
<tr>
<td>STRMSGSZ (valid for HP-UX 11iv1 only)</td>
<td>65535</td>
</tr>
</tbody>
</table>
6. Select **Actions > Process New Kernel**.
7. Click **Yes** on the information window to confirm your decision to restart the machine.
8. Follow on-screen instructions to restart your machine and to enable the new settings.

**Parent topic:** Setting up a stand-alone server on HP-UX

**Next topic:** Installing WebSphere Portal on HP-UX
Installing WebSphere Portal on HP-UX

IBM® WebSphere® Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

Review the Supported hardware and software requirements for your operating system and ensure that all hardware and software matches the minimum product level before installing WebSphere Portal. Review Installation methods, options, and sources before installing to determine your installation source location and installation method. Note: If you plan to use IBM WebSphere Application Server Version 7.0, you must purchase and install it before installing WebSphere Portal on the existing WebSphere Application Server Version 7.0 instance.

- Prerequisites
  - Preparing your HP-UX operating system

Perform the following steps to install WebSphere Portal:

1. Type ping yourserver.yourcompany.com on a command line to verify that your fully qualified host name is properly configured.
2. Type ping localhost on a command line to verify that your network is properly configured.
3. If you are installing with an existing WebSphere Application Server instance, ensure it is installed at the supported level and has the following features installed:
   - Restriction: Installing WebSphere Portal Version 6.1.0.1 or higher on an existing WebSphere Application Server Version 7.0 instance where WebSphere Portal Version 7.0 was installed is unsupported.
   - Important: Besides ensuring that the existing WebSphere Application Server instance is at the supported level, you will also need to ensure that Apache Derby is installed at the supported level before installing WebSphere Portal. See WebSphere Portal detailed system requirements for supported levels.

   - Application Server
   - Administration
     - Scripted Administration
     - Administrative Console
   - Ant and Deployment Tools
     - Deploy Tool
     - Ant Utilities

4. Optional: Perform the following steps if you want to install WebSphere Portal using a non-root user:
   - Restriction: Although it is possible to install WebSphere Application Server as a non-root user, there are limitations to this option that can affect WebSphere Portal functionality, which is why you should install WebSphere Application Server as a root user.
   - A. Install the current supported version of WebSphere Application Server as a root user.
   - B. Open a command prompt and perform the following steps to create a non-root user and to change ownership:

      1. Use the appropriate system commands to create a new group, to create a new user, and to add the new user to the new group.
2. Run the following tasks to change the rights of the non-root user:

   chmod -R g+rwX /opt/IBM
   chgrp -R group_name /opt/IBM
   chmod -R g+wr /tmp
   chgrp -R group_name /tmp
   chmod -R g+wr /var/tmp
   chgrp -R group_name /var/tmp

C. Launch the installation program per the next step. During the installation, select the existing WebSphere Application Server using the non-root user and set the installation location to a unique location under the path where non-root permissions were granted.

5. Choose one of the following installation commands based on the installation method you want to use to install WebSphere Portal; see Installation Methods for more information:

   **Advance installation parameters:** To customize your installation, you can add various parameters to your installation commands; for example, you can specify your port information. See Advanced installation parameters for information.

   **Restriction:** When defining your installation path, the **ONLY** supported characters are ASCII letters [A-Z and a-z], numbers [0-9], slashes [/ or \ depending on your operating system], and the dash [-].

   **Table 1. Installation task options**

<table>
<thead>
<tr>
<th>Installation type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphical user interface</td>
<td>./install.sh</td>
</tr>
<tr>
<td>Console mode</td>
<td>./install.sh -console from the PortalExpress subdirectory</td>
</tr>
<tr>
<td>Silent Install</td>
<td>./install.sh -options &quot;path_to_file/response_filename&quot;, where path_to_file is the full path to the response file and response_filename is the name of the file. A sample install response file (installresponse.txt) and a sample uninstall response file (uninstallresponse.txt) are located in the setup CD root directory. <strong>Important:</strong> Do not place the response file in a path that contains a space and do not put a space in the file name.</td>
</tr>
</tbody>
</table>

   **Note:** If the installation program does not detect a WebSphere Application Server instance that you know exists, exit the installation program and pass the WebSphere Application Server instance location using the command line; for example, ./install.sh -W was.undetectedWas="/my/WAS/location".

   **Note:** If the GUI or console mode installation program fails to detect ports for either WebSphere Application Server or WebSphere Portal, a warning message displays and the installer offers another chance to enter the values. If the silent installation fails to detect ports for either WebSphere Application Server or WebSphere Portal, the installer will exit.

6. Verify your installation was successful; access WebSphere Portal using the http://yourserver:yourport/wps/portal format. Run the ./ConfigEngine.sh list-server-ports -DWasPassword=password task from the wp_profile_root /ConfigEngine directory to generate the wp_profile_root/ConfigEngine/log/wp_PortMatrix.txt file that lists WebSphere Portal ports for your installation.

7. Optional: After the WebSphere Portal installation completes successfully, run the ./ConfigEngine.sh configure-express -DPortalAdminPwd=password -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, if you want to enable the sample IBM Lotus Web Content Management content, such as internet and intranet sample sites.

   **Restriction:** Run the configure-express task before configuring your database, user registry, context root, security, etc. If you ran any tasks other than the install task, do not run this task.

   **Note:** See http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_overview.html
for tutorials on how to use the sample content.

The sample content includes:

- Creates a group called contentAuthors; members of this group are given privileges to create content in the sample Internet and intranet sites. Navigate to the Administration area and then click Access > Users and Groups.

- Creates two new Web Content Management Libraries: “Internet Web Content 6.1.0” and “Intranet Web Content 6.1.0”.

- Adds a portlet filter and applies the filter to various portlets in the sample Internet and intranet sites. You can see the definition of the filter in the WebSphere Application Server Administration console and examining the custom resources under the Environment area.

- Creates two new theme policies: InternetStyle and IntranetStyle. These styles are applied to sample Internet and intranet sites. Navigate to Theme Customizer and then select the style.

- Creates several portlet clones of the Web Content Management rendering portlet. These portlet clones are used on sample Internet and intranet sites.

- Creates two virtual portals with context roots of wps/portal/intranet and wps/portal/internet. These are the sample Internet and intranet sites. Go to http://yourserver:yourport/wps/portal/internet and http://yourserver:yourport/wps/portal/intranet to access them.

- Creates several sample credential slots, including "Default slot for E-mail", "Default slot for Feeds", "Default slot for Miscellaneous", "Default slot for Web Clipping", and "Default slot for Web Content Management". Navigate to the Administration area and then click Access > Credential Vault > Manage System Vault Slots.

- Modifies the portlet palette to include some new portlets. Click the Portlets link to see the portlet palette.

- Optional: If you ran the configure-express task, the owner of the items in the Web content libraries containing the Internet and Intranet Site Template content will be listed as uid=xyzadmin,o=defaultWIMFileBasedRealm. To update the owner information for these items to correspond to your portal administrator ID, complete the following steps:
  
  A. Edit the wp_profile_root

     /PortalServer/wcm/shared/app/config/wcm-services/MemberFixerModule.properties file.

  B. Add the following line to the file:

     uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN

     Replace portal_admin_DN with the distinguished name of the portal administrator.

  C. Save your changes and close the file.

  D. Run the ./ConfigEngine.sh action-express-memberfixer -DPortalAdminPwd=password -DWasPassword=password task, located in the wp_profile_root/ConfigEngine directory.

9. Optional: Perform the following steps if you need to change the ports for WebSphere Application Server or WebSphere Portal:

   Note: The starting port parameter is required for a successful completion of the modify-ports-by-startport task.

   Once you specify a start port, this port becomes the base for assigning port values. The code increments this value as each port is assigned, which means that the WebSphere Portal ports will be assigned incrementally starting with the port defined with the modify-ports-by-startport task.

   A. Stop the server1 and WebSphere_Portal servers.

   B. Run one of the following commands for each server you need to change:

<table>
<thead>
<tr>
<th>Method</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting port number</td>
<td>./ConfigEngine.sh modify-ports-by-startport -</td>
</tr>
<tr>
<td></td>
<td>DWasPassword=password -</td>
</tr>
<tr>
<td></td>
<td>DModifyPortsServer=servername -</td>
</tr>
<tr>
<td></td>
<td>DStartPort=starting port number -</td>
</tr>
</tbody>
</table>
### Restart the server1 and WebSphere_Portal servers.

1. Log in to the WebSphere Application Server Administration Console.
2. Click **Servers > Server Types > WebSphere application servers > WebSphere_Portal**.
4. In the **Generic JVM arguments** field, change the **MaxPermSize** value to `-XX:MaxPermSize=numerator value`, where `numerator value` is a quarter of the value entered for the **Maximum Heap Size**. **Important:** If **MaxPermSize** does not exist in the **Generic JVM arguments** field, add it to the field but do not replace existing information in the **Generic JVM arguments** field with the **MaxPermSize** information.
5. Click **OK** to save your changes.
6. Click **Save** to save your changes to the master configuration.
7. Log out of the WebSphere Application Server Administration Console.
8. Restart the server1 and WebSphere_Portal servers.

---

**Port file** **Note:** Sample port files are available on the Setup disc.

```sh
./ConfigEngine.sh modify-ports-by-portsfile
DWasPassword=password
DModifyPortsServer=servername
DPortsFile=full path to ports file
```

The following is an example of the information within a port file although the port values will be different based on your environment:

- `BOOTSTRAP_ADDRESS=10031`
- `SOAP_CONNECTOR_ADDRESS=10032`
- `SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=10032`
- `CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=10025`
- `CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=10036`
- `WC_adminhost=10027`
- `WC_defaulthost=33344`
- `DCSunicast_address=10029`
- `WC_adminhost_secure=10039`
- `WC_defaulthost_secure=10035`
- `SIB_ENDPOINT_ADDRESS=10026`
- `SIB_ENDPOINT_SECURE_ADDRESS=10037`
- `SIB_MQ_ENDPOINT_ADDRESS=10030`
- `SIB_MQ_ENDPOINT_SECURE_ADDRESS=10028`
- `ORB_LISTENER_ADDRESS=10034`

---

**Parent topic:** Setting up a stand-alone server on HP-UX  
**Previous topic:** Preparing your HP-UX operating system  
**Next topic:** Configuring databases

### Related concepts
- Installation methods
- IBM Support Assistant Lite for WebSphere Portal

### Related tasks
- Creating and maintaining multiple profiles on HP-UX  
Using the member fixer tool with IBM Lotus Web Content Management

### Related reference
- Sample port files are available on the Setup disc.
Advanced installation parameters
Configuring databases

Before transferring default data to the production database server, you need to prepare the database server. Database preparation includes creating user IDs and databases. For this installation path, stand-alone production server, remote databases servers are used.

- **Prerequisites**
  - Preparing your HP-UX operating system
  - Installing WebSphere Portal on HP-UX

Select the database server that are using.

- **Preparing DB2 on AIX for a stand-alone production server**
  You can install DB2 on the same server as WebSphere Portal, also referred to as local, or on a different server, referred to as remote. If you plan to use a local DB2 database, scripts are provided to simplify the database creation and transfer. More information about the local database is provide in the single server installation instructions. For a remote database you must complete the transfer manually.

- **Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server**
  You can install DB2 on the same server as WebSphere Portal, also referred to as local, or on a different server, referred to as remote. If you plan to use a local DB2 database, scripts are provided to simplify the database creation and transfer. More information about the local database is provided in the single server installation instructions. For a remote database you must complete the transfer manually.

- **Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server**
  Setting up the Oracle database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

- **Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server**
  Setting up the Oracle RAC database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

- **Setting up a remote SQL Server database on Windows for a stand-alone production server**
  To setup a remote Microsoft SQL Server Enterprise Edition database, create user IDs and databases on a remote server. Tasks are provided to assist with creating the user IDs and the databases. Before you can use the tasks, you must modify properties files.

- **Verifying databases**
  After you configure IBM WebSphere Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.

Parent topic: Setting up a stand-alone server on HP-UX
Previous topic: Installing WebSphere Portal on HP-UX
Next topic: Preparing a remote Web server when portal is installed on HP-UX
Preparing DB2 on AIX for a stand-alone production server

You can install DB2 on the same server as WebSphere Portal, also referred to as local, or on a different server, referred to as remote. If you plan to use a local DB2 database, scripts are provided to simplify the database creation and transfer. More information about the local database is provide in the single server installation instructions. For a remote database you must complete the transfer manually.

1. **Installing DB2**
   View information on installing DB2 for use with WebSphere Portal.

2. **Create users**
   View the steps to set up users for DB2 to work with WebSphere Portal.

3. **Creating remote databases**
   A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

4. **Optional: Assigning custom table spaces**
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

5. **Modifying database properties**
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

6. **Optional: Configuring JCR collation support**
   View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

7. **Configure WebSphere Portal to use DB2**
   View information on how to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

8. **Configuring DB2 for large file handling in Web Content Management**
   If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

9. **Optional: Changing driver types**
   WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Parent topic: Configuring databases
Installing DB2

View information on installing DB2 for use with WebSphere Portal.

Before you begin:

- Make sure DB2 is set up with updated kernel parameters according to the recommendations in the DB2 Quick Beginnings guide at DB2 Technical Support.
- **(For Type 2 drivers only):** DB2 Connect must be installed on the same system as WebSphere Portal, and have the same name as the server profile name.
- When you install DB2 using the DB2 installation program, it automatically creates a DB2 administrative user with the correct operating system rights.

1. If you are using the JDBC driver in type 2 mode, configure your DB2 Connect client with the following commands. If you are using a remote database, complete this step separately from the WebSphere Portal installation.
   - `db2 update dbm cfg using tp_mon_name WAS`
   - `db2 update dbm cfg using spm_name hostname`, where `hostname` is the host name of WebSphere Portal.
     Because the default for `spm_name` is the hostname itself, specifying the `hostname` parameter is optional. If your hostname is more than eight characters, use empty double quotes (" "). For example, `db2 update dbm cfg using spm_name " "`.

2. Before installing DB2, log in with a user ID that has administrative authority.

3. To install DB2, follow the instructions that are provided with the DB2 documentation to install DB2 and the required fix pack.

**Parent topic:** Preparing DB2 on AIX for a stand-alone production server

**Next topic:** Create users
Create users

View the steps to set up users for DB2 to work with WebSphere Portal.

Before you begin: You should have completed Installing DB2.

- **Prerequisites**
  - Installing DB2

For all databases, use one user with administrative rights on the operating system and the DB2 installation. This user can be the database administrative user that is created automatically by the DB2 installation program. This user is the database configuration user that will be used for configuration tasks: creating database tables and performing database transfer. If you choose to have WebSphere Portal also create the databases, the database user should be given **SYSADM** rights. You only need to manually create an administrator ID when you do not want to use an existing DB2 administrator ID.

A common user name is `db2inst1`, but you can assign any user name as long as it has administrative access and follows the limitations listed here. Do not change the user name after creating it.

The limitations on user names are:
- User names can contain one to eight characters.
- Group and instance names can contain one to eight characters.
- Names cannot be any of the following: `users` `admins` `guests` `public` `local`
- Names cannot begin with: `IBM` `SQL` `SYS`
- Names cannot include accented characters.
- Create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Following are the basic steps to create new users and grant privileges. For more information, refer to the DB2 product documentation.

1. From the DB2 Control Center, expand the object tree until you find the DB Users folder.
2. Right-click the DB Users folder and select **Create/Add**. The Create DB User notebook opens.
3. In the User name field, type a user ID.
4. Assign an authority for the user. .
5. Click **OK**.

**Parent topic:** Preparing DB2 on AIX for a stand-alone production server

**Previous topic:** Installing DB2

**Next topic:** Creating remote databases
Creating remote databases

A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

Before creating these databases, note the following information:

- Use IBM® DB2 Universal Database™ Enterprise Server Edition JDBC driver, JCC, type 4.
- If the DB2 JDBC type 4 driver is used, only the instructions on the remote server are required. You do not need to install the DB2 client software, and you do not need to complete the steps related to the DB2 client.
- The client software, DB2 Connect, must be correctly configured to connect to the remote DB2 server instance, for example, `db2inst1`.
- These instructions assume that a remote DB2 server and DB2 Connect are already installed and running. For information on installing DB2 Connect, see the DB2 product documentation.

**Prerequisites**

- Installing DB2
- Create users

To create a database, you must be a DB2 System Administrator with sufficient database privileges (SYSADM or at a minimum SYSCTRL).

1. Log in to the DB2 server system as a user with sufficient database privileges.
2. Initialize a DB2 command environment by opening a command prompt and typing `su - db2inst1`, where `db2inst1` is the database user. The prompt changes to your operating system shell prompt, for example: `$`. In this mode, you must type `db2` at the beginning of each command and use double quotation marks (""") around the entire command. The following example shows how to use the double quotes; it is not an actual command that you run:

   ```
   db2 "CREATE REGULAR TABLESPACE SMS PAGESIZE 4 K MANAGED BY SYSTEM USING ('sms') BUFFERPOOL SMSPOOL"
   ```

   **Note:** If you are using the Command Line Processor (CLP), refer to your DB2 documentation for details. The command prompt is `db2=>`. In this mode, commands can be entered without the `db2` prefix or the double quotation marks. However, the following steps assume you are not using the CLP and are entering commands from your operating system shell prompt, for example: `$`.

3. Run the following commands on the DB2 server system to configure the DB2 database instance:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Description</th>
</tr>
</thead>
</table>
| **DB2**     | db2set DB2_RR_TO_RS=YES
             | DB2_EVALUNCOMMITTED=YES
             | db2set DB2_INLIST_TO_NLJN=YES
             | db2 "UPDATE DBM CFG USING query_heap_sz 32768"
             | db2 "UPDATE DBM CFG USING maxagents 500"
             | db2 "UPDATE DBM CFG USING sheapthres 0" |

4. **(Important)** Failure to complete this step will result in an unsuccessful database transfer. Run the following commands on the DB2 server system to create the necessary databases:

   **Notes:**
   - Replace `dbname` with the actual name of the database. Run the commands and each time replace `dbname` with the actual values for release, community, customization, Java Content Repository, Feedback, and Likeminds. You will need to run the commands once for each database for a total of six times.
   - **Remember:** DB2 database names cannot...
exceed eight characters. Therefore, consider using these database names: release, commun, custom, jcrdb, fdbkdb, and lmdb.

```
db2 "CREATE DB dbname using codeset UTF-8 territory us PAGESIZE 8192"
db2 "UPDATE DB CFG FOR dbname USING applheapsz 4096"
db2 "UPDATE DB CFG FOR dbname USING app_ctl_heap_sz 1024"
db2 "UPDATE DB CFG FOR dbname USING stmtheap 32768"
db2 "UPDATE DB CFG FOR dbname USING dbheap 2400"
db2 "UPDATE DB CFG FOR dbname USING locklist 1000"
db2 "UPDATE DB CFG FOR dbname USING logfilsiz 4000"
db2 "UPDATE DB CFG FOR dbname USING logprimary 12"
db2 "UPDATE DB CFG FOR dbname USING logsecond 20"
db2 "UPDATE DB CFG FOR dbname USING logbufsz 32"
db2 "UPDATE DB CFG FOR dbname USING avg_appls 5"
db2 "UPDATE DB CFG FOR dbname using AUTO_MAINT off"
```

5. Complete the following:

A. On the DB2 server system, run the following commands. This step is only required for the IBM Java Content Repository database (jcrdb).

- `jcrdb` is the name of the database used to store user data and objects
- `jcr` is the jcr user for `jcrdb` *Note:* This value can be replaced with any ID that has administrative authority.
- `dbpassword` is the password for the jcr user for the `jcrdb`

```
db2 "CONNECT TO jcrdb USER jcr USING dbpassword"
db2 "CREATE BUFFERPOOL ICMLSFREQBP4 SIZE 1000 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL ICMLSVOLATILEBP4 SIZE 16000 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL ICMLSMAINBP32 SIZE 16000 PAGESIZE 32 K"
db2 "CREATE BUFFERPOOL CMBSMAIN4 SIZE 1000 PAGESIZE 4 K"
db2 "CREATE REGULAR TABLESPACE ICMLFQ32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLFQ32') BUFFERPOOL ICMLSMAINBP32"  
```

B. For JDBC Type 2 connections only: On DB2 Connect, use a text editor to open the `/etc/services` file. If it does not specify the DB2 connection service port, add the following text to specify the port for the remote DB2 instance:

```
db2inst1port1/tcp # DB2 connection service port
```

where `db2inst1` is the name of the DB2 instance ID on the system, and `port1` with the actual port number that is
assigned to the DB2 connection service in your DB2 server installation. The connection service port on the DB2
Client system, WebSphere Portal server, must match the connection service port on the DB2 server. The ports
should match by number but not necessarily by name.

C. Set up the correct service name by entering the following command on the DB2 server system: `db2 "UPDATE DBM
CFG USING svcename svce_name"` where `svce_name` is the connection service port name that is specified above.

D. For JDBC Type 2 connections only: On DB2 Connect, set DB2COMM to TCP/IP by using the `db2set` command
`db2set DB2COMM=tcpip`.

E. For JDBC Type 2 connections only: Catalog the TCP/IP node with the IP address of the remote database server on
DB2 Connect: `db2 "catalog tcpip node remote_db_node_alias remote database_server_node server
connection_service_port"` where:
- `remote_db_node_alias` is the alias name of the database server that you are defining for the WebSphere Application
Server node name. The alias name can contain one to eight characters.
- `database_server_node` is the fully qualified host name of your database server system.
- `connection_service_port` is the name of the DB2 connection service port that is configured in the `/etc/services
file on the database server system.

F. For JDBC Type 2 connections only, catalog the WebSphere Portal databases on DB2 Connect, where:
- `remote_db_name_domain`, is the cataloged name of the databases on the server system for each domain.
- `domain_alias_name`, is the database alias names that you are defining.
- `remote_db_node_alias` is the name that was used previously when you cataloged the TCP/IP node in the previous
step.

The alias for each database must be different from the actual database name and can only contain up to eight
characters. `db2 "catalog db remote_db_name_release as release_alias_name at node remote_db_node_alias"

`db2 "catalog db remote_db_name_community as comm_alias_name at node remote_db_node_alias"

`db2 "catalog db remote_db_name_customization as cust_alias_name at node remote_db_node_alias"

`db2 "catalog db remote_db_name_fdbkdb as fdbkdb_alias_name at node remote_db_node_alias"

`db2 "catalog db remote_db_name_lmdb as lmdb_alias_name at node remote_db_node_alias"

`db2 "catalog db remote_db_name_jcrdb as jcrdb_alias_name at node remote_db_node_alias"

G. For JDBC Type 2 connections only: Log out of DB2 Connect by entering `db2 "terminate"`.

H. For JDBC Type 2 connections only, on DB2 Connect, test your remote connection by issuing the following command
in the DB2 command window: `db2 "connect to alias_name user username using password", where alias_name
is the alias name that you defined above, username is the database user, and password is the password assigned to
the database user.

Parent topic: Preparing DB2 on AIX for a stand-alone production server

Previous topic: Create users

Next topic: Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- The page size of table spaces used by WebSphere Portal must be 8192 bytes.
- For details on creating table spaces refer to the documentation for the database.

**Prerequisites**
- Installing DB2
- Create users
- Creating remote databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`

   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - `release`
   - `community`
   - `customization`
   - `jcr`
   - `feedback`
   - `likeminds`

3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword `IN` and a space. For example: `community.COMP_INST.tablespace=IN COMM8KSPACE`

   Repeat this step for each domain that you are transferring.

4. Save and close `dbdomain.space_mapping.properties`.

5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - Windows: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
   - UNIX: `.ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

**Parent topic:** Preparing DB2 on AIX for a stand-alone production server

**Previous topic:** Creating remote databases

**Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

Tip: For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Working with properties files:
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:
  - release.DbName=release
  - jcr.DbName=jcrdb
  - feedback.DbName=fdbkdb
  - likeminds.DbName=lmdb
  - community.DbName=commdb
  - customization.DbName=custdb
- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.
   A. For dbdomainDbType, type db2.
   B. For dbdomainDbName, type the name of the WebSphere Portal domain database. Notes:
      - This value is also the database element in the dbdomainDbUrl property.
      - This value is the TCP-IP alias for the database.
   C. For dbdomainDbSchema, type the schema name of the database domain. Note: Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   D. For dbdomainDataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For dbdomainDbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. Note: The database element of this value should match the value of DbName.
   F. For dbdomainDbUser, type the user ID for the database administrator.
   G. For dbdomainDbPassword, type the password for the database administrator.
   H. Optional: For dbdomainDbRuntimeUser, type the user ID of the database user that should be used by WebSphere Portal to connect to the database at runtime.
   I. If dbdomainDbRuntimeUser is specified, you must set dbdomainDbRuntimePassword to be the password of the runtime database user.
   J. For dbdomainDBADbUser, type the database administrator user ID for privileged access operations during creation of the database.
K. For **dbdomain.DBA.DbPassword**, type the database administrator password for privileged access operations during creation of the database.

L. For **dbdomain.XDbName**, type the database loop back alias that needs to be set if you plan to use the `create-database` task. **Note:** Required only for local databases using a JDBC Type 2 driver.

M. For **dbdomain.DbNode**, type the value for the node database. Set this value if you want to call `create-database`. **Note:** Required only for local databases.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

   A. For **source.domainDbType**, type of the database you are currently configured to use. The value for **source.domain.DbType** is Derby by default.

   B. For **source.domain.DbName**, type the name of the database domain you are currently using.

   C. For **source.domain.DbSchema**, type current schema identifier for objects within the database for this domain.

   D. For **source.domain.DataSourceName**, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

   E. For **source.domain.DbUrl**, type the url currently used to access your database.

   F. For **source.domain.DbUser**, type the name of the user accessing this database.

   G. For **source.domain.DbPassword**, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

   A. For **db2.DbDriver**, type the name of the JDBC driver class.

   B. For **db2.DbLibrary**, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For **db2.JdbcProviderName**, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

   A. For **WasPassword**, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

---

**Parent topic:** Preparing DB2 on AIX for a stand-alone production server

**Previous topic:** Assigning custom table spaces

**Next topic:** Configuring JCR collation support
Configuring JCR collation support

View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties

1. Stop the WebSphere Portal server:
   ```
   .\stopServer.sh WebSphere_Portal -username admin_userid -password admin_password
   ```

2. If using a remote DB2 server, copy the following files from the WebSphere Portal server to a temporary directory on the remote DB2 server:
   ```
   PortalServer_root/jcr/prereq.jcr/config/collation.jar
   PortalServer_root/jcr/prereq.jcr/config/registerCollationUDFTemplate.sql
   ```

3. Set up collation on the database where the JCR domain is located.
   A. Change to the directory `db2_instance_owner_home/sqlib/function`.
   B. Execute the command:
      ```
      Remote DB2: db2_instance_owner_home/sqlib/java/jdk/bin/jar -xvf temporary location/collation.jar icm/CollationUDF.class
      Local DB2: db2_instance_owner_home/sqlib/java/jdk/bin/jar -xvf PortalServer_root/jcr/prereq.jcr/config/collation.jar icm/CollationUDF.class
      ```
   C. If using a local DB2, change to the directory `wp_profile_root/PortalServer/jcr/config`. If using a remote DB2, change to the temporary directory where you copied the files in a previous step.
   D. Open the file `registerCollationUDFTemplate.sql` and change all `SCHEMA` references to the JCR schema; for example, `JCR`. The value set for `SCHEMA` should match the value set for the `jcr.DbSchema` property. You specify `jcr.DbSchema` in the configuration file `wkplc_comp.properties` when you modify database properties.

4. Connect to the JCR database by running `db2 connect to <jcrdb> user <userid> using <password>`. You must connect to the JCR database using the userid and password of the WebSphere Portal JCR data source.

5. Edit the `icm.properties` file, located in `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm` directory. Add the following section to the end of the file:
   ```
   # Enable/Disable collation support for all DB2 platforms
   ```
# Disabled by default
jcr.query.collation.db2.enabled = true

# Database specific collation mappings
# These mappings apply map a Java locale name into a collation name
# supported by the underlying database.
# Example mappings for DB2 platform

# English
jcr.query.collation.en = en

# Swedish
jcr.query.collation.sv = sv

jcr.query.collation.zh = zh
jcr.query.collation.de = de
jcr.query.collation.da = da
jcr.query.collation.hu = hu
jcr.query.collation.jp = jp

6. Enter the following command to start the WebSphere Portal server:

../startServer.sh WebSphere_Portal
Configure WebSphere Portal to use DB2

View information on how to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties

Tips:
- To run these tasks as a non-root user, you must first run the task chown -R non-root_user WebSphereDir.
- If you are transferring from Oracle, the `open_cursors` setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

Steps for transferring data to another supported database

1. If you are running a type 2 connection, before transferring data edit the `db2cli.ini` file. Failure to follow these steps will cause the database transfer to hang at the task action-process-constraints.
   A. Locate the file `/home/db2inst1/sqllib/cfg/db2cli.ini`
   B. Add the following to the end of the file. Leave an empty line after `ReturnAliases=0.[COMMON]
      
          [COMMON]
          DYNAMIC=1
          ReturnAliases=0

2. Perform this step only if you are installing multiple instances of WebSphere Portal. Change the maximum number of databases `MAX_NETBIOS_CONNECTIONS` to increase the default configured number of databases. For example, enter the following command at the database prompt: `set client MAX_NETBIOS_CONNECTIONS 254` A message indicates success if the number was increased.

3. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.

4. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2. .
   `/home/db2inst1/sqllib/db2profile`
   where `db2inst1` represents your database instance
   Note: You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.

5. Enter the following commands to validate configuration properties...
   `/ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password`
./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password

6. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.

7. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td>./stopServer.sh server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

8. Transfer the database:

**Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

A. Change to the directory wp_profile_root/ConfigEngine.

B. Enter the following command: ./ConfigEngine.sh database-transfer -DWasPassword=password

**Note:**
- To select specific database domains to transfer, modify the -DTransferDomainList specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command: ./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
- If you have been storing data in IBM® Derby for a long time, database transfer could fail with OutOfMemory exceptions. If database transfer fails, add the following property to the command in this step: ./ConfigEngine.sh database-transfer -DDbtJavaMaxMemory=1536M -DWasPassword=password

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files and then repeat this step.

9. After transferring the database tables, perform a reorg check to improve performance. Perform this step for each database alias in the property file.

A. Connect to a database with the following command:
db2 connect to database_alias user db2admin_userid using password

**Note:** Additional options might be required if additional security has been installed. Refer to DB2 Universal Database commands by example for links to the command reference.

B. After it is connected, run the following commands from the DB2 prompt:
db2 reorgchk update statistics on table all > xyz.out

C. Look in the reorg column for entries marked with an * (star or asterisk) in the file xyz.out. For each line with an *, note the tablename and run the following command for each tablename:
db2 reorg table tablename db2 terminate db2rbind database_name -l db2rbind.out -u db2_admin -p password

D. The output file db2rbind.out is only created when there is an error for the db2rbind command.

10. Change to the directory wp_profile_root/bin.

11. Enter the following command to start the WebSphere Portal server: ./startServer.sh WebSphere_Portal

---

**Parent topic:** Preparing DB2 on AIX for a stand-alone production server

**Previous topic:** Configuring JCR collation support

**Next topic:** Configuring DB2 for large file handling in Web Content Management
Configuring DB2 for large file handling in Web Content Management

If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties
  - Configure WebSphere Portal to use DB2

**Note:** You only need to perform these steps if you are using Web Content Management.

1. Log into the WebSphere Application Server administrative console.
2. Click **Resources > JDBC > Data sources**.
3. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
4. Select the name of the data source that is defined in `wkplc_comp.properties` for the JCR database domain. The default data source is `wpdbDS`.
5. Click **Custom properties**.
6. Ensure that the `fullyMaterializeLobData` property is set to `false`.

Parent topic: Preparing DB2 on AIX for a stand-alone production server

Previous topic: Configure WebSphere Portal to use DB2

Next topic: Changing driver types
Changing driver types

WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Before you begin, ensure that the following conditions are met:

- The WebSphere Portal database has been successfully transferred to DB2 using the database-transfer configuration task.
- The files wkplc_comp.properties and wkplc_dbtype.properties have been modified to set the correct values for the DB2 drivers that you are switching to:

  In the file wkplc_comp.properties set each <Domain>.DbUrl property using the following formats:

  - db2 (type 2): { jdbc:db2:wpsdb }
  - db2 (type 4): { jdbc:db2://<YourDatabaseServer>:50000/wpsdb:returnAlias=0; }

  In the file wkplc_dbtype.properties set the db2.DbLibrary property using the following format:

  - For DB2 Type 2 driver use <SQLLIB>/java/db2java.zip
  - For DB2 Type 4 driver use <SQLLIB>/java/db2jcc.jar:<SQLLIB>/java/db2jcc_license_cu.jar

  In the file wkplc_dbtype.properties set the db2.DbDriver property using the following format:

  - For DB2 Type 2 driver use COM.ibm.db2.jdbc.app.DB2Driver
  - For DB2 Type 4 driver use com.ibm.db2.jcc.DB2Driver

**Note:** If WebSphere Portal is installed on the same machine as the DB2 server and you switch from a JDBC Type 4 connection to a JDBC Type 2 connection, verify that you have created the alias names for the DB2 databases as described in Creating remote databases and that the alias names are specified for the databases in the file wkplc_comp.properties.

When switching from a JDBC Type 2 connection to a JDBC Type 4 connection, remove the database alias names and refer to the databases directly. This is required because of a limitation in the DB2 Universal JDBC driver.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties
  - Configure WebSphere Portal to use DB2
  - Configuring DB2 for large file handling in Web Content Management

1. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.

2. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2.

   ```bash
   /home/db2inst1/sqllib/db2profile
   ```

   where `db2inst1` represents your database instance

   **Note:** You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.

3. Enter the following commands to validate configuration properties:

   ```bash
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ```
4. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td><code>./stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td><code>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

6. Change to the directory `wp_profile_root/ConfigEngine`.

7. To change from one supported driver to the other, run the following task to connect the database, including only the domains that require the switch:

   ```
   ```

8. Change to the directory `wp_profile_root/bin`.

9. Enter the following command to start the WebSphere Portal server:

   ```
   ./startServer.sh WebSphere_Portal
   ```

#### Parent topic:
Preparing DB2 on AIX for a stand-alone production server

#### Previous topic:
Configuring DB2 for large file handling in Web Content Management
Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

You can install DB2 on the same server as WebSphere Portal, also referred to as local, or on a different server, referred to as remote. If you plan to use a local DB2 database, scripts are provided to simplify the database creation and transfer. More information about the local database is provided in the single server installation instructions. For a remote database you must complete the transfer manually.

1. **Installing DB2**
   View information on installing DB2 for use with WebSphere Portal.

2. **Create users**
   View the steps to set up users for DB2 to work with WebSphere Portal.

3. **Creating remote databases**
   A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

4. **Optional: Assigning custom table spaces**
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

5. **Modifying database properties**
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

6. **Optional: Configuring JCR collation support**
   View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

7. **Configure WebSphere Portal to use DB2**
   View the steps to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

8. **Configuring DB2 for large file handling in Web Content Management**
   If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

9. **Optional: Changing driver types**
   WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

**Parent topic:** Configuring databases
Installing DB2

View information on installing DB2 for use with WebSphere Portal.

Before you begin:

- Make sure DB2 is set up with updated kernel parameters according to the recommendations in the DB2 Quick Beginnings guide at DB2 Technical Support.
- **(For Type 2 drivers only):** DB2 Connect must be installed on the same system as WebSphere Portal, and have the same name as the server profile name.
- When you install DB2 using the DB2 installation program, it automatically creates a DB2 administrative user with the correct operating system rights.

1. If you are using the JDBC driver in type 2 mode, configure your DB2 Connect client with the following commands. If you are using a remote database, complete this step separately from the WebSphere Portal installation.
   - `db2 update dbm cfg using tp_mon_name WAS`
   - `db2 update dbm cfg using spm_name hostname`, where `hostname` is the host name of WebSphere Portal.
   Because the default for `spm_name` is the hostname itself, specifying the `hostname` parameter is optional. If your hostname is more than eight characters, use empty double quotes (" "). For example, `db2 update dbm cfg using spm_name " "`.
2. Before installing DB2, log in with a user ID that has administrative authority. The system user ID and password must match the database user ID and password for the WebSphere Portal databases. If you do not use the default DB2 database user ID, or if you must access a remote database, create the system user ID before installing DB2.

   To set up the environment for the database access:

   A. The initialization script for this user (for example, `user-home/.profile`) must contain a call to the `db2profile` script in the `db-home/sqllib` directory.

   B. After you create the system user ID and password for the DB2 installation, add the user ID to the DB2 administration group (such as `db2admin`) for that system.

   To ensure that the user is set up correctly, log in with this user ID and start the DB2 command line processor by executing the command: `db2`. If the command line processor displays, the user ID is set up correctly.

3. To install DB2, follow the instructions that are provided with the DB2 documentation to install DB2 and the required fix pack.

Parent topic: Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

Next topic: Create users
Create users

View the steps to set up users for DB2 to work with WebSphere Portal.

Before you begin: You should have completed Installing DB2.

- Prerequisites
  - Installing DB2

For all databases, use one user with administrative rights on the operating system and the DB2 installation. This user can be the database administrative user that is created automatically by the DB2 installation program. This user is the database configuration user that will be used for configuration tasks: creating database tables and performing database transfer. If you choose to have WebSphere Portal also create the databases, the database user should be given **SYSADM** rights. You only need to manually create an administrator ID when you do not want to use an existing DB2 administrator ID.

A common user name is `db2inst1`, but you can assign any user name as long as it has administrative access and follows the limitations listed here. Do not change the user name after creating it.

The limitations on user names are:

- User names can contain one to eight characters.
- Group and instance names can contain one to eight characters.
- Names cannot be any of the following: `users`, `admins`, `guests`, `public`, `local`
- Names cannot begin with: `IBM`, `SQL`, `SYS`
- Names cannot include accented characters.
- Create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Following are the basic steps to create new users and grant privileges. For more information, refer to the DB2 product documentation.

1. From the DB2 Control Center, expand the object tree until you find the DB Users folder.
2. Right-click the DB Users folder and select **Create/Add**. The Create DB User notebook opens.
3. In the User name field, type a user ID.
4. Assign an authority for the user.
5. Click **OK**.

**Parent topic:** Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Installing DB2

**Next topic:** Creating remote databases
Creating remote databases

A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

Before creating these databases, note the following information:

- Use IBM® DB2 Universal Database™ Enterprise Server Edition JDBC driver, JCC, type 4.
- If the DB2 JDBC type 4 driver is used, only the instructions on the remote server are required. You do not need to install the DB2 client software, and you do not need to complete the steps related to the DB2 client.
- The client software, DB2 Connect, must be correctly configured to connect to the remote DB2 server instance, for example, `db2inst1`.
- These instructions assume that a remote DB2 server and DB2 Connect are already installed and running. For information on installing DB2 Connect, see the DB2 product documentation.

**Prerequisites**

- Installing DB2
- Create users

To create a database, you must be a DB2 System Administrator with sufficient database privileges (SYSADM or at a minimum SYSCTRL).

1. Log in to the DB2 server system as a user with sufficient database privileges.
2. Initialize a DB2 command environment by opening a command prompt and typing `su - db2inst1`, where `db2inst1` is the database user. The prompt changes to your operating system shell prompt, for example: `$`. In this mode, you must type `db2` at the beginning of each command and use double quotation marks (""") around the entire command. The following example shows how to use the double quotes; it is not an actual command that you run:

   ```sh
db2 "CREATE REGULAR TABLESPACE SMS PAGESIZE 4 K MANAGED BY SYSTEM USING ('sms') BUFFERPOOL SMSPOOL"
   ```

   **Note:** If you are using the Command Line Processor (CLP), refer to your DB2 documentation for details. The command prompt is `db2=>`. In this mode, commands can be entered without the `db2` prefix or the double quotation marks. However, the following steps assume you are not using the CLP and are entering commands from your operating system shell prompt, for example: `$`.
3. Run the following commands on the DB2 server system to configure the DB2 database instance:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>db2set DB2_RA_TO_RS=YES</code></td>
</tr>
<tr>
<td></td>
<td><code>db2set DB2_EVALUNCOMMITTED=YES</code></td>
</tr>
<tr>
<td></td>
<td><code>db2set DB2_INLIST_TO_NLJN=YES</code></td>
</tr>
<tr>
<td></td>
<td><code>db2 &quot;UPDATE DBM CFG USING query_heap_sz 32768&quot;</code></td>
</tr>
<tr>
<td></td>
<td><code>db2 &quot;UPDATE DBM CFG USING maxagents 500&quot;</code></td>
</tr>
<tr>
<td></td>
<td><code>db2 &quot;UPDATE DBM CFG USING sheapthres 0&quot;</code></td>
</tr>
</tbody>
</table>

4. **(Important)** Failure to complete this step will result in an unsuccessful database transfer. Run the following commands on the DB2 server system to create the necessary databases:

   **Notes:**
   - Replace `dbname` with the actual name of the database. Run the commands and each time replace `dbname` with the actual values for release, community, customization, Java Content Repository, Feedback, and Likeminds. You will need to run the commands once for each database for a total of six times. **Remember:** DB2 database names cannot
exceed eight characters. Therefore, consider using these database names: release, commun, custom, jcrdb, fdbkdb, and lmdb.

db2 "CREATE DB dbname using codeset UTF-8 territory us PAGESIZE 8192"
db2 "UPDATE DB CFG FOR dbname USING applheapsz 4096"
db2 "UPDATE DB CFG FOR dbname USING app_ctl_heap_sz 1024"
db2 "UPDATE DB CFG FOR dbname USING stmtheap 32768"
db2 "UPDATE DB CFG FOR dbname USING dbheap 2400"
db2 "UPDATE DB CFG FOR dbname USING locklist 1000"
db2 "UPDATE DB CFG FOR dbname USING logfilsiz 4000"
db2 "UPDATE DB CFG FOR dbname USING logprimary 12"
db2 "UPDATE DB CFG FOR dbname USING logsecond 20"
db2 "UPDATE DB CFG FOR dbname USING logbufsz 32"
db2 "UPDATE DB CFG FOR dbname USING avg_appls 5"
db2 "UPDATE DB CFG FOR dbname using AUTO_MAINT off"

Complete the following:

5. On the DB2 server system, run the following commands. This step is only required for the IBM Java Content Repository database (jcrdb).

- jcrdb is the name of the database used to store user data and objects
- jcr is the jcr user for jcrdb
- dbpassword is the password for the jcr user for the jcrdb

For JDBC Type 2 connections only: On DB2 Connect, use a text editor to open the /etc/services file. If it does not specify the DB2 connection service port, add the following text to specify the port for the remote DB2 instance:

```
db2c_db2inst1 port1 /tcp # DB2 connection service port
```

where db2inst1 is the name of the DB2 instance ID on the system, and port1 with the actual port number that is
assigned to the DB2 connection service in your DB2 server installation. The connection service port on the DB2 Client system, WebSphere Portal server, must match the connection service port on the DB2 server. The ports should match by number but not necessarily by name.

C. Set up the correct service name by entering the following command on the DB2 server system: 

```
db2 "UPDATE DBM CFG USING svcname svce_name" where svce_name is the connection service port name that is specified above.
```

D. For JDBC Type 2 connections only: On DB2 Connect, set DB2COMM to TCP/IP by using the db2set command

```
db2set DB2COMM=tcpip.
```

E. For JDBC Type 2 connections only: Catalog the TCP/IP node with the IP address of the remote database server on DB2 Connect:

```
db2 "catalog tcpip node remote_db_node_alias remote database_server_node server connection_service_port" where:
- remote_db_node_alias is the alias name of the database server that you are defining for the WebSphere Application Server node name. The alias name can contain one to eight characters.
- database_server_node is the fully qualified host name of your database server system.
- connection_service_port is the name of the DB2 connection service port that is configured in the /etc/services file on the database server system.
```

F. For JDBC Type 2 connections only, catalog the WebSphere Portal databases on DB2 Connect, where:

- remote_db_name_domain, is the cataloged name of the databases on the server system for each domain.
- domain_alias_name, is the database alias names that you are defining.
- remote_db_node_alias is the name that was used previously when you cataloged the TCP/IP node in the previous step.

The alias for each database must be different from the actual database name and can only contain up to eight characters.

```
db2 "catalog db remote_db_name_release as release_alias_name at node remote_db_node_alias"
db2 "catalog db remote_db_name_community as comm_alias_name at node remote_db_node_alias"
db2 "catalog db remote_db_name_customization as cust_alias_name at node remote_db_node_alias"
db2 "catalog db remote_db_name_fdbkdb as fdbkdb_alias_name at node remote_db_node_alias"
db2 "catalog db remote_db_name_lmdb as lmdb_alias_name at node remote_db_node_alias"
db2 "catalog db remote_db_name_jcrdb as jcrdb_alias_name at node remote_db_node_alias"
```

G. For JDBC Type 2 connections only: Log out of DB2 Connect by entering 

```
db2 "terminate".
```

H. For JDBC Type 2 connections only, on DB2 Connect, test your remote connection by issuing the following command in the DB2 command window: 

```
db2 "connect to alias_name user username using password", where alias_name is the alias name that you defined above, username is the database user, and password is the password assigned to the database user.
```

Parent topic: Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Create users

Next topic: Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:

- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the wp_profile_root
  /PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- The page size of table spaces used by WebSphere Portal must be 8192 bytes.
- For details on creating table spaces refer to the documentation for the database.

Prerequisites

- Installing DB2
- Create users
- Creating remote databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.index_name.indexspace
   For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
     - release
     - community
     - customization
     - jcr
     - feedback
     - likeminds
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword IN and a space. For example: community.COMP_INST.tablespace=IN COMM8KSPACE
   Repeat this step for each domain that you are transferring.
4. Save and close dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
   - UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

**Parent topic:** Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server  
**Previous topic:** Creating remote databases  
**Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

Tip: For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Working with properties files:
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release(DbName), jcr(DbName), feedback(DbName), and likeminds(DbName). For example:
  - release(DbName)=release
  - jcr(DbName)=jcrdb
  - feedback(DbName)=fdbkdb
  - likeminds(DbName)=lmdb
  - community(DbName)=commdb
  - customization(DbName)=custdb

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds

- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.
   A. For dbdomainDbType, type db2.
   B. For dbdomainDbName, type the name of the WebSphere Portal domain database. Notes:
      - This value is also the database element in the dbdomainDbUrl property.
      - This value is the TCP-IP alias for the database.
   C. For dbdomainDbSchema, type the schema name of the database domain. Note: Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   D. For dbdomainDataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For dbdomainDbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. Note: The database element of this value should match the value of DbName.
   F. For dbdomainDbUser, type the user ID for the database administrator.
   G. For dbdomainDbPassword, type the password for the database administrator.
   H. Optional: For dbdomainDbRuntimeUser, type the user ID of the database user that should be used by WebSphere Portal to connect to the database at runtime.
   I. If dbdomainDbRuntimeUser is specified, you must set dbdomainDbRuntimePassword to be the password of the runtime database user.
   J. For dbdomainDBADBA, type the database administrator user ID for privileged access operations during creation of the database.
K. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.

L. For `dbdomain.XDbName`, type the database loop back alias that needs to be set if you plan to use the `create-database` task. **Note:** Required only for local databases using a JDBC Type 2 driver.

M. For `dbdomain.DbNode`, type the value for the node database. Set this value if you want to call `create-database`. **Note:** Required only for local databases.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

   A. For `source.domainDbType`, type of the database you are currently configured to use. The value for `source.domainDbType` is Derby by default.

   B. For `source.domainDbName`, type the name of the database domain you are currently using.

   C. For `source.domainDbSchema`, type current schema identifier for objects within the database for this domain.

   D. For `source.domainDataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

   E. For `source.domainDbUrl`, type the url currently used to access your database.

   F. For `source.domainDbUser`, type the name of the user accessing this database.

   G. For `source.domainDbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

   A. For `db2.DbDriver`, type the name of the JDBC driver class.

   B. For `db2.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For `db2.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server  
**Previous topic:** Assigning custom table spaces  
**Next topic:** Configuring JCR collation support
**Configuring JCR collation support**

View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

### Prerequisites
- Installing DB2
- Create users
- Creating remote databases
- Modifying database properties

1. Stop the WebSphere Portal server:
   ```
   ./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password
   ```

2. If using a remote DB2 server, copy the following files from the WebSphere Portal server to a temporary directory on the remote DB2 server:
   ```
   PortalServer_root/jcr/prereq.jcr/config/collation.jar
   PortalServer_root/jcr/prereq.jcr/config/registerCollationUDFTemplate.sql
   ```

3. Set up collation on the database where the JCR domain is located.
   A. Change to the directory `db2_instance_owner_home/sqllib/function`.
   B. Execute the command:
      - **Remote DB2**:
        ```
        db2_instance_owner_home/sqllib/java/jdk/bin/jar -xvf temporary location/collation.jar icm/CollationUDF.class
        ```
      - **Local DB2**:
        ```
        db2_instance_owner_home/sqllib/java/jdk/bin/jar -xvf PortalServer_root/jcr/prereq.jcr/config/collation.jar icm/CollationUDF.class
        ```
   C. If using a local DB2, change to the directory `wp_profile_root/PortalServer/jcr/config`. If using a remote DB2, change to the temporary directory where you copied the files in a previous step.

   D. Open the file `registerCollationUDFTemplate.sql` and change all `SCHEMA` references to the JCR schema; for example, `JCR`. The value set for `SCHEMA` should match the value set for the `jcr.DbSchema` property. You specify `jcr.DbSchema` in the configuration file `wkplc_comp.properties` when you modify database properties.

   E. Connect to the JCR database by running `db2 connect to <jcrdb> user <userid> using <password>`. You must connect to the JCR database using the userid and password of the WebSphere Portal JCR data source.

   F. Execute the following command. **Remote DB2**:
      ```
      db2 -tvf temporary location/registerCollationUDFTemplate.sql
      ```
   G. Disconnect from the JCR database.

4. Verify that the UDF is registered properly.
   A. Log in as the `db2instanceID`. Open a DB2 terminal window, and type `db2`. From the command line, type the command `connect to JCRDB user userid using password`. You must connect to the JCR database using the userid and password of the WebSphere Portal JCR data source.
   B. Execute the following command in the previous substep. If the command completes successfully, the UDF is registered correctly as follows: `values schema.sortkeyj('abc','en')`, where `schema` is the schema used in the previous substep.

5. Edit the `icm.properties` file, located in `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm` directory. Add the following section to the end of the file:
   ```
   # Enable/Disable collation support for all DB2 platforms
   ```

---

**Operating systems:** AIX, HP-UX, i5/OS, Linux, Solaris, Windows

**Portal, Version 6.1.5**
# Disabled by default
jcr.query.collation.db2.enabled = true

# Database specific collation mappings
# These mappings apply map a Java locale name into a collation name
# supported by the underlying database.
# Example mappings for DB2 platform

# English
jcr.query.collation.en = en

# Swedish
jcr.query.collation sv = sv

jcr.query.collation.zh = zh
jcr.query.collation.de = de
jcr.query.collation.da = da
jcr.query.collation.hu = hu
jcr.query.collation.jp = jp

6. Enter the following command to start the WebSphere Portal server:

```bash
./startServer.sh WebSphere_Portal
```

**Parent topic:** Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server  
**Previous topic:** Modifying database properties  
**Next topic:** Configure WebSphere Portal to use DB2
Configure WebSphere Portal to use DB2

View the steps to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties

Tips:
- To run these tasks as a non-root user, you must first run the task chown -R non_root_user WebSphereDir.
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

Steps for transferring data to another supported database

1. If you are running a type 2 connection, before transferring data edit the db2cli.ini file. Failure to follow these steps will cause the database transfer to hang at the task action-process-constraints.
   A. Locate the file /home/db2inst1/sqllib/cfg/db2cli.ini.
   B. Add the following to the end of the file. Leave an empty line after ReturnAliases=0.[COMMON]
      DYNAMIC=1
      ReturnAliases=0

2. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.

3. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2. .
   /home/db2inst1/sqllib/db2profile
   where db2inst1 represents your database instance
   Note: You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.

4. Enter the following commands to validate configuration properties...
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password

5. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.
6. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td><code>.stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td><code>.stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

7. Transfer the database:

**Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

A. Change to the directory `wp_profile_root/ConfigEngine`.

B. Enter the following command: `.ConfigEngine.sh database-transfer -DWasPassword=password`

**Note:**
- To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command: `.ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password`
- If you have been storing data in IBM® Derby for a long time, database transfer could fail with OutOfMemory exceptions. If database transfer fails, add the following property to the command in this step: `.ConfigEngine.sh database-transfer -DDbtJavaMaxMemory=1536M -DWasPassword=password`

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

8. After transferring the database tables, perform a reorg check to improve performance. Perform this step for each database alias in the property file.

A. Connect to a database with the following command:

```
db2 connect to database_alias user db2admin_userid using password
```

**Note:** Additional options might be required if additional security has been installed. Refer to DB2 Universal Database commands by example for links to the command reference.

B. After it is connected, run the following commands from the DB2 prompt:

```
db2 reorgchk update statistics on table all > xyz.out
```

C. Look in the reorg column for entries marked with a * (star or asterisk) in the file `xyz.out`. For each line with a *, note the `tablename` and run the following command for each `tablename`:

```
db2 reorg table tablename db2 terminate db2rbind database_name -l db2rbind.out -u db2_admin -p password
```

D. The output file `db2rbind.out` is only created when there is an error for the `db2rbind` command.

9. Change to the directory `wp_profile_root/bin`.

10. Enter the following command to start the WebSphere Portal server:

```
./startServer.sh WebSphere_Portal
```

Parent topic: Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Configuring JCR collation support

Next topic: Configuring DB2 for large file handling in Web Content Management
Configuring DB2 for large file handling in Web Content Management

If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

**Prerequisites**
- Installing DB2
- Create users
- Creating remote databases
- Modifying database properties
- Configure WebSphere Portal to use DB2

**Note:** You only need to perform these steps if you are using Web Content Management.

1. Log into the WebSphere Application Server administrative console.
2. Click **Resources > JDBC > Data sources**.
3. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
4. Select the name of the data source that is defined in `wkplc_comp.properties` for the JCR database domain. The default data source is `wpdbDS`.
5. Click **Custom properties**.
6. Ensure that the `fullyMaterializeLobData` property is set to `false`.

**Parent topic:** Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Configure WebSphere Portal to use DB2

**Next topic:** Changing driver types
Changing driver types

WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Before you begin, ensure that the following conditions are met:

- The WebSphere Portal database has been successfully transferred to DB2 using the database-transfer configuration task.
- The files wkplc_comp.properties and wkplc_dbtype.properties have been modified to set the correct values for the DB2 drivers that you are switching to:

In the file wkplc_comp.properties set each <Domain>.DbUrl property using the following formats:

```
# db2 (type 2):        { jdbc:db2:wpsdb }
# db2 (type 4):        { jdbc:db2://<YourDatabaseServer>:50000/wpsdb:returnAlias=0; }
```

In the file wkplc_dbtype.properties set the db2.DbLibrary property using the following format:

```
# For DB2 Type 2
driver use <SQLLIB>/java/db2java.zip
# For DB2 Type 4 driver use <SQLLIB>/java/db2jcc.jar:<SQLLIB>/java/db2jcc_license_cu.jar
```

In the file wkplc_dbtype.properties set the db2.DbDriver property using the following format:

```
# For DB2 Type 2
driver use COM.ibm.db2.jdbc.app.DB2Driver
# For DB2 Type 4 driver use com.ibm.db2.jcc.DB2Driver
```

**Note:** If WebSphere Portal is installed on the same machine as the DB2 server and you switch from a JDBC Type 4 connection to a JDBC Type 2 connection, verify that you have created the alias names for the DB2 databases as described in Creating remote databases and that the alias names are specified for the databases in the file wkplc_comp.properties. When switching from a JDBC Type 2 connection to a JDBC Type 4 connection, remove the database alias names and refer to the databases directly. This is required because of a limitation in the DB2 Universal JDBC driver.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties
  - Configure WebSphere Portal to use DB2
  - Configuring DB2 for large file handling in Web Content Management

1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.
2. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2.

   ```
   /home/db2inst1/sqlib/db2profile
   where db2inst1 represents your database instance
   **Note:** You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.
   ```

3. Enter the following commands to validate configuration properties.

   ```
   .\ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ```
4. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.
5. Stop both WebSphere Application Server and the WebSphere Portal server:

```
Option                                Description
WebSphere Application Server          ./stopServer.sh server1 -username admin_userid -password admin_password
WebSphere Portal                      ./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password
```

6. Change to the directory `wp_profile_root/ConfigEngine`.
7. To change from one supported driver to the other, run the following task to connect the database, including only the domains that require the switch:

```
```
8. Change to the directory `wp_profile_root/bin`.
9. Enter the following command to start the WebSphere Portal server:

```
./startServer.sh WebSphere_Portal
```

Parent topic: Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Configuring DB2 for large file handling in Web Content Management
Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Setting up the Oracle database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

1. Installing Oracle
   View information on how to install Oracle for use with WebSphere Portal.

2. Creating databases
   View some important considerations before setting up Oracle databases to work with WebSphere Portal.

3. Creating users
   View the steps to set up users for Oracle to work with WebSphere Portal.

4. Modifying database properties
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

6. HP-UX stand-alone: Creating JCR table spaces
   This topic provides manual instructions for creating JCR table spaces for Oracle.

7. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

8. Configure WebSphere Portal to use Oracle
   This section provides information on how to manually transfer data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configuring databases
Installing Oracle

View information on how to install Oracle for use with WebSphere Portal.

You can use Oracle as the database software. You must install Oracle separately and before performing a database transfer.

Refer to Oracle product documentation for installation instructions.

**Parent topic:** Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server  
**Next topic:** Creating databases
Creating databases

View some important considerations before setting up Oracle databases to work with WebSphere Portal.

For information about creating databases, refer to the Oracle product documentation. For information on the recommended database architecture and the databases you will need to create, see the Planning for Oracle topic. Be sure that all databases to be used with WebSphere Portal are created as UNICODE character set databases.

If you are using remote Oracle 9 or 10 databases, you must also copy the ojdbc14.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM® WebSphere® Application Server Version 7.0, you must also copy the ojdbc6.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM WebSphere Application Server Version 6.1, you must also copy the ojdbc5.jar file from the remote Oracle server to the WebSphere Portal machine. The typical location is the oracle_home/sqldeveloper/jdbc/lib directory. Record the copy location on your local machine for future reference.

When creating Oracle databases for use with WebSphere Portal, you should consider the following information:

- The Oracle databases must be created manually before configuring WebSphere Portal.
- All databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle server will be remote or local to the WebSphere Portal installation.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle database in order for WebSphere Portal to communicate with the Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes:
  - db_block_size = 8192 bytes
  - db_cache_size = 307,200 bytes
  - db_files = 1024 files
  - log_buffer = 65536 bytes
  - open_cursors = 1500 cursors
  - pga_aggregate_target = 204,800 bytes
  - pre_page_sga = true
  - processes = 300 processes
  - shared_pool_size = 204,800 bytes

  **Note:** If you are using IBM Java Content Repository, the open_cursors value may need to be increased based on the table count in the Java Content Repository schema.

- Raise the number of parallel servers as appropriate. For example, if you have more than 875 parallel servers, you should set the parallel_max_servers to 1200.
- The Oracle parameter CURSOR_SHARING allows similar SQL Statements to be shared when possible, which prevents parsing and establishing a new execution plan. The execution plan is used by Oracle to gather the data needed to satisfy a request. There are two options for CURSOR_SHARING, which are as follows:
  - **FORCE**
    - When you select this option, Oracle uses the same execution plan for all SQLs that are similar in value even if the values are different. When you use this option, the execution plan may not provide optimum performance. For example, similar SQLs with different values may behave differently when executed running the same plan.
  - **EXACT**
- When you select this option, Oracle only shares the same execution plan for SQLs that are identical and use the same values. This option removes the risk of a SQL statement being executed when optimum performance conditions do not exist.

WebSphere Portal supports both options. Regardless of the option selected, portlet applications should not be affected. Contact your database administrator for further assistance on these options.

**- Prerequisites**

  - Installing Oracle

Parent topic: Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Installing Oracle

Next topic: Creating users
Creating users

View the steps to set up users for Oracle to work with WebSphere Portal.

Before you begin:
- You should have installed Oracle.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

Prerequisites
- Installing Oracle
- Creating databases

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Start the Oracle SQL command line processor ‘SQL*Plus’ by entering `sqlplus /nolog` on the operating system command prompt
2. Log in to the Oracle database by executing command `connect user/password@database name`. For example: `system/manager@wpsdb` will log the administrative user system with a password of manager into the wpsdb database.
3. Create the WebSphere Portal user `ddomain.DbUser`, where `ddomain` is replaced by release, community, or customization.
   A. Log in to the database in which you want to create the new users.
   B. Create the users that you need in the appropriate databases by entering the following commands in the SQL*Plus tool, where `user_tablespace` is the default tablespace that is identified by the database administrator to store user objects and `temp_tablespace` is identified to store temporary objects. All names must be in lower case.

```
SQL> create user releasusr identified by password
default tablespace user_tablespace
temporary tablespace temp_tablespace;
GRANT UNLIMITED TABLESPACE TO releasusr;
SQL> create user communityusr identified by password
default tablespace user_tablespace temporary tablespace temp_tablespace;
GRANT UNLIMITED TABLESPACE TO communityusr;
SQL> create user customizationusr identified by password
default tablespace user_tablespace
temporary tablespace temp_tablespace;
GRANT UNLIMITED TABLESPACE TO customizationusr;
SQL> create user jcr identified by password
default tablespace user_tablespace
temporary tablespace temp_tablespace;
GRANT UNLIMITED TABLESPACE TO jcr
```
4. Connect to the Feedback database:
   A. Enter the following command: `SQL> connect`
B. Enter `user-name: username/password@dbname` where `username` is an existing administrative user in the database. For example, `system/manager@fdbkdb` will log the administrative user system with a password of manager into the fdbkdb database.

C. Create the Feedback user:

```sql
SQL> create user feedback identified by password
default tablespace user_tablespace
temporary tablespace temp_tablespace;
GRANT UNLIMITED TABLESPACE TO feedback
```

D. Log out of the command line tool using the command `SQL> exit`.

5. Connect to the LikeMinds database:

A. Enter the following command:

```sql
SQL> connect
```

B. Enter `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example, `system/manager@lmdb` will log the administrative user system with a password of manager into the lmdb database.

C. Create the LikeMinds user:

```sql
SQL> create user lmdbusr identified by password
default tablespace user_tablespace
temporary tablespace temp_tablespace;
GRANT UNLIMITED TABLESPACE TO lmdbusr
```

D. Log out of the command line tool using the command `SQL> exit`.

6. Perform the following steps to verify that you can connect to the database with the user ID and password that you just created:

A. Enter the `SQL> connect` command to connect to the content database.

B. Enter `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example, `system/manager@jcrdb` will log the administrative user system with a password of manager into the Oracle database.

---

Parent topic: Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server  
Previous topic: Creating databases  
Next topic: Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- Prerequisites
  - Installing Oracle
  - Creating databases
  - Creating users

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

Tip: For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Working with properties files:
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:
  - release.DbName=release
  - jcr.DbName=jcrdb
  - feedback.DbName=fdbkdb
  - likeminds.DbName=lmdb
  - community.DbName=commdb
  - customization.DbName=customdb
- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.
   A. For dbdomainDbType, type oracle.
   B. For dbdomainDbName, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the dbdomainDbUrl property.
   C. For dbdomainDbSchema, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The dbdomainDbName should be the same value used for the dbdomainDbSchema
   
   **Restriction:** The value for dbdomainDbSchema must equal the value for dbdomainDbUser.
   D. For dbdomainDataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For dbdomainDbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:**
      - The database element of this value should match the value of DbName.
      - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following:
        jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.
F. For dbdomain.DbUser, type the user ID for the database administrator. **Restriction:** The value for dbdomain .DbUser must equal the value for dbdomain.DbSchema.

G. For dbdomain.DbPassword, type the password for the database administrator.

H. For dbdomain.DBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.

I. For dbdomain.DBA.DbPassword, type the database administrator password for privileged access operations during creation of the database.

J. For dbdomain.DbHome, type the root location for the database. **Note:** This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For source.domain.DbType, type of the database you are currently configured to use. The value for source.domain .DbType is Derby by default.

B. For source.domain.DbName, type the name of the database domain you are currently using.

C. For source.domain.DbSchema, type current schema identifier for objects within the database for this domain.

D. For source.domain.DataSourceName, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For source.domain.DbUrl, type the url currently used to access your database.

F. For source.domain.DbUser, type the name of the user accessing this database.

G. For source.domain.DbPassword, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file wkplc_dbtype.properties.

   A. For oracle.DbDriver, type the name of the class that SqlProcessor uses to import SQL files.

   B. For oracle.DbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For oracle.JdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file wkplc.properties.

   A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Creating users

**Next topic:** Setting up databases

---

|||

---
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

- **Prerequisites**
  - Installing Oracle
  - Creating databases
  - Creating users
  - Modifying database properties

1. On the database server, make sure that the folder `db_1` contains the subfolders `your_oracle_instance/data` and `your_oracle_instance/index`. If this folder hierarchy does not exist, create it manually before you run the `setup-database` task. The `setup-database` task requires these folders to create database users. If these folders do not exist, the `setup-database` task will fail.

2. Change to the directory `wp_profile_root/ConfigEngine`

3. To create the database users, type the following command:

```
./ConfigEngine.sh setup-database -DWasPassword=password
```

**Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:

`<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql`.

**Parent topic:** Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Modifying database properties

**Next topic:** HP-UX stand-alone: Creating JCR table spaces
HP-UX stand-alone: Creating JCR table spaces

This topic provides manual instructions for creating JCR table spaces for Oracle.

- **Prerequisites**
  - Installing Oracle
  - Creating databases
  - Creating users
  - Modifying database properties
  - Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.
2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:
   - `&jcrdb` is the name of the database you created to store user data.
   - `&dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.

   Ensure that the `.` is included in the variables when you substitute the values of your environment with these variables.

   **Important:** You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

   ```sql
   create tablespace ICMLFQ32 datafile '&dbpath./&jcrdb./data/&jcrdb._ICMLFQ32_01.dbf' size 300M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLNF32 datafile '&dbpath./&jcrdb./data/&jcrdb._ICMLNF32_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMVFQ04 datafile '&dbpath./&jcrdb./data/&jcrdb._ICMVFQ04_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMSFQ04 datafile '&dbpath./&jcrdb./data/&jcrdb._ICMSFQ04_01.dbf' size 150M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLSNDX datafile '&dbpath./&jcrdb./index/&jcrdb._ICMLSNDX_01.dbf' size 10M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   ```

   A. Set the size, autoextend, and maxsize values according to your environment. For example, you may want to change the maxsize to a set value rather than UNLIMITED.

   B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.

   C. Refer to the Oracle command reference for more information about using the `create tablespace` command.

**Parent topic:** Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Setting up databases

**Next topic:** Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:

- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root`/PortalServer/config/tablesspaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

**Prerequisites**

- Installing Oracle
- Creating databases
- Creating users
- Modifying database properties
- Setting up databases
- HP-UX stand-alone: Creating JCR table spaces

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tablesspaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`

   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds

3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword `TABLESPACE` and a space. For example: `community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE`

   Repeat this step for each domain that you are transferring.

4. Save and close `dbdomain.space_mapping.properties`.

5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - Windows: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
- UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

Parent topic: Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server
Previous topic: HP-UX stand-alone: Creating JCR table spaces
Next topic: Configure WebSphere Portal to use Oracle
Configure WebSphere Portal to use Oracle

This section provides information on how to manually transfer data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- Prerequisites
  - Installing Oracle
  - Creating databases
  - Creating users
  - Modifying database properties
  - Setting up databases
  - HP-UX stand-alone: Creating JCR table spaces

Tips:
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

Steps for transferring data to another supported database

1. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.
2. Enter the following commands to validate configuration properties.

   ```bash
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=\password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=\password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=\password
   ```

3. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.
4. Stop both WebSphere Application Server and the WebSphere Portal server:

   ```bash
   ./stopServer.sh server1 -username admin_userid -password admin_password
   ```
5. Transfer the database. **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

A. Change to the directory `wp_profile_root/ConfigEngine`.

B. Enter the following command:

```
./ConfigEngine.sh database-transfer -DWasPass=pass
```

**Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:

```
./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPass=password
```

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example:

```
SQL> execute dbms_stats.gather_schema_stats(ownname=>'jcr', cascade=>TRUE);
```

7. Change to the directory `wp_profile_root/bin`.

8. Enter the following command to start the WebSphere Portal server:

```
./startServer.sh WebSphere_Portal
```

**Parent topic:** Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Assigning custom table spaces
Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Setting up the Oracle RAC database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

1. Installing Oracle RAC
   View information on how to install Oracle RAC for use with WebSphere Portal.

2. Create users
   View the steps to set up users for Oracle RAC to work with WebSphere Portal.

3. Creating databases
   View some important considerations before setting up Oracle RAC databases to work with WebSphere Portal.

4. Modifying database properties
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

6. HP-UX stand-alone: Creating JCR table spaces
   This topic provides manual instructions for creating JCR table spaces for Oracle.

7. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

8. Configuring WebSphere Portal to use Oracle RAC
   This section provides information on how to manually transfer data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configuring databases
Installing Oracle RAC

View information on how to install Oracle RAC for use with WebSphere Portal.

Before you begin:
- You should have completed reviewing the Planning for Oracle RAC section.
- Ensure the Oracle CRS (Cluster Ready Service) and Oracle RAC databases have been installed and configured on the primary and secondary nodes.
- Run the following commands on both RAC nodes to start global services daemon (GSD), oracle listeners, and agents.
  - `$ gsdctl start`
  - `$ lsnrctl start`
  - `$ agentctl start`

You can use Oracle RAC as the database software for WebSphere Portal. You must install Oracle RAC separately and before performing WebSphere Portal database transfer.

Refer to Oracle product documentation for installation instructions.

Parent topic: Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Next topic: Create users
Create users

View the steps to set up users for Oracle RAC to work with WebSphere Portal.

Before you begin:
- You should have completed installing Oracle RAC.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

- **Prerequisites**
  - Installing Oracle RAC

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Create the users in the WebSphere Portal databases
   A. Log in to the database in which you want to create the new users.
   B. Create the users that you need in the appropriate databases by entering the following command in the SQL*Plus tool, where \texttt{user\_tablespace} is the default tablespace that is identified by the database administrator to store user objects and \texttt{temp\_tablespace} is identified to store temporary objects: \textbf{Tip}: Balance the storage of user objects among tablespaces to prevent running out of space with overuse of \texttt{user\_tablespace}. Also consider increasing the size of \texttt{user\_tablespace} when handling a high volume of users.

   \begin{verbatim}
   SQL> create user username identified by password
default tablespace user_tablespace
temporary tablespace temp_tablespace;
   \end{verbatim}

2. Log in by entering the command \texttt{sqlplus} in SQL*Plus:

3. Enter the command \texttt{user-name: username/password@dbname}, where \texttt{username} is an existing administrative user in the database. For example: system/manager@wpsdb will log the administrative user system with a password of manager into the wpsdb database.

4. Create the WebSphere Portal user \texttt{dbdomain.DbUser}, where \texttt{dbdomain} is replaced by release, community, customization, or JCR. All names must be in lower case. The following code uses the default user tablespace and the default temporary tablespace. Explicitly specify these tablespaces if you do not want to use the defaults.

   \begin{verbatim}
   SQL> create user releaseusr identified by password;
   SQL> create user communityusr identified by password;
   SQL> create user customizationusr identified by password;
   SQL> create user jcr identified by password;
   \end{verbatim}

5. Connect to the content database by entering the command \texttt{sqlplus} in SQL*Plus.

6. Enter the following, where \texttt{username} is an existing administrative user in the database: \texttt{user-name: username/password@dbname} For example, system/manager@jcrdb will log the administrative user system with a password of manager into the Oracle RAC database.

7. Create the Oracle RAC users by entering the following: When creating the \texttt{jcr} user, grant all necessary privileges. If you do not grant privileges, you will receive the error \texttt{ICMADMIN lacks CREATE SESSION Privilege logon denied}
when you try to connect with the jcr user:

```
SQL> create user jcr identified by password default tablespace users temporary tablespace temp;
```

8. Connect to the Feedback database:

A. Enter the following command: `SQL> connect`

B. Enter `user-name: username/password@dbname` where `username` is an existing administrative user in the database.

For example, `system/manager@fdbkdb` will log the administrative user `system` with a password of `manager` into the `fdbkdb` database.

9. Create the Feedback user:

```
SQL> create user feedback identified by password default tablespace users temporary tablespace temp;
```

10. Connect to the LikeMinds database:

```
SQL> connect
```

11. Enter `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example, `system/manager@lmdb` will log the administrative user `system` with a password of `manager` into the `lmdb` database.

12. Create the LikeMinds user:

```
SQL> create user lmdbusr identified by password default tablespace users temporary tablespace temp;
```

13. Log out of the command line tool using the command `SQL> exit`.

**Parent topic:** Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Installing Oracle RAC

**Next topic:** Creating databases
Creating databases

View some important considerations before setting up Oracle RAC databases to work with WebSphere Portal.

When creating Oracle RAC databases for use with WebSphere Portal, you should consider the following information:

- The Oracle RAC databases must be created manually before configuring WebSphere Portal.
- All databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle RAC server will be remote or local to the WebSphere Portal installation.
- If using an earlier version of Oracle RAC (9i or earlier), ensure that Oracle RAC JVM is also installed.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle RAC database in order for WebSphere Portal to communicate with the IBM Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes are: 

  \[
  \begin{align*}
  \text{db_block_size} &= 8192 \text{ bytes} \\
  \text{db_cache_size} &= 314,572,800 \text{ bytes} \\
  \text{log_buffer} &= 65536 \text{ bytes} \\
  \text{open_cursors} &= 1500 \text{ cursors} \\
  \text{pga_aggregate_target} &= 209,715,200 \text{ bytes} \\
  \text{pre_page_sga} &= \text{true} \\
  \text{processes} &= 300 \text{ processes} \\
  \text{shared_pool_size} &= 209,715,200 \text{ bytes}
  \end{align*}
  \]

  \textbf{Note:} If you are using IBM Java Content Repository, the \texttt{open_cursors} value may need to be increased based on the table count in the IBM Java Content Repository schema.

- \textbf{Prerequisites}
  - Installing Oracle RAC
  - Create users

Refer to the following instructions to create tablespaces:

1. In the database directory, create the data directory \texttt{data} and the index directory \texttt{index}.
2. Create the tablespaces. \textbf{Important:} You must use the same tablespace names listed in the commands. The tablespace names cannot be customized or modified.
   
   A. Find and edit the SQL script \texttt{jcr_ora_tablespaces.sql} in the directory \texttt{wp_profile_root}/ConfigEngine/work/db/oracle.
   
   B. In the define section, replace the following variables with the values from your environment:

   - \texttt{jcrdb} - Name of the database you created to store user data.
   - \texttt{logfile} - Location to store the log file.
   - \texttt{dbpath} - Directory where you created the database.

   C. Set the size, autoextend, and maxsize values according to your environment. For example, you may want to change the maxsize to a set value rather than UNLIMITED. Consult your Database Administrator for more info.
D. Execute the following SQL script: **Notes:**
- The DBA or a user with sufficient credentials (CREATE tablespace) must execute the script.
- The script will prompt for the database username and password.

```
# sqlplus
SQL > @jcr_ora_tablespaces.sql
```

**Parent topic:** Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Create users

**Next topic:** Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Installing Oracle RAC
  - Create users
  - Creating databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:
  - `release.DbName` = `release`
  - `jcr.DbName` = `jcrdb`
  - `feedback.DbName` = `fdbkdb`
  - `likeminds.DbName` = `lmdb`
  - `community.DbName` = `commdb`
  - `customization.DbName` = `custdb`
- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace `dbdomain` with:
  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
- \textit{dbdomain}.DbType
- \textit{dbdomain}.DbName
- \textit{dbdomain}.DbUrl
- \textit{dbdomain}.DbSchema

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If \textit{DbUser}, \textit{DbUrl}, and \textit{DbPassword} are not the same across domains, the value for \textit{DataSourceName} must differ from the \textit{DataSourceName} of the other domains. In other words, this value must be unique for the database domain.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

1. Locate the following files and create a backup copy of each before changing any values:
   - \texttt{wp_profile_root}/ConfigEngine/properties/wkplc.properties
   - \texttt{wp_profile_root}/ConfigEngine/properties/wkplc_comp.properties
   - \texttt{wp_profile_root}/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In \texttt{wkplc_comp.properties}, most properties are repeated for each domain.

2. Use a text editor to open the properties file \texttt{wkplc_comp.properties} and modify the values to correspond to your environment.
   A. For \textit{dbdomain}.DbType, type \texttt{oracle}.
   B. For \textit{dbdomain}.DbName, type the name of the WebSphere Portal domain database. \textbf{Note:} This value is also the database element in the \textit{dbdomain}.DbUrl property.
   C. For \textit{dbdomain}.DbSchema, type the schema name of the database domain. \textbf{Note:} Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The \textit{dbdomain}.DbName should be the same value used for the \textit{dbdomain}.DbSchema
      \begin{center}
      \textbf{Restriction:} The value for \textit{dbdomain}.DbSchema must equal the value for \textit{dbdomain}.DbUser.
      \end{center}
   D. For \textit{dbdomain}.DataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For \textit{dbdomain}.DbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. \textbf{Note:}
      - The database element of this value should match the value of DbName.
      - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following:
        \texttt{jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME}. \textbf{When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.}
For `dbdomain.DbUser`, type the user ID for the database administrator. **Restriction:** The value for `dbdomain.DbUser` must equal the value for `dbdomain.DbSchema`.

G. For `dbdomain.DbPassword`, type the password for the database administrator.

H. For `dbdomain.DBA.DbUser`, type the database administrator user ID for privileged access operations during creation of the database.

I. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.

J. For `dbdomain.DbHome`, type the root location for the database. **Note:** This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For `source.domain.DbType`, type of the database you are currently configured to use. The value for `source.domain.DbType` is Derby by default.

B. For `source.domain.DbName`, type the name of the database domain you are currently using.

C. For `source.domain.DbSchema`, type current schema identifier for objects within the database for this domain.

D. For `source.domain.DataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For `source.domain.DbUrl`, type the url currently used to access your database.

F. For `source.domain.DbUser`, type the name of the user accessing this database.

G. For `source.domain.DbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

   A. For `oracle.DbDriver`, type the name of the class that SqlProcessor uses to import SQL files.

   B. For `oracle.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For `oracle.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Creating databases

**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

Before you begin, ensure that the following prerequisites are met:
- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

- **Prerequisites**
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties

1. On the database server, make sure that the folder `db_1` contains the subfolders `your_oracle_instance/data` and `your_oracle_instance/index`. If this folder hierarchy does not exist, create it manually before you run the `setup-database` task. The `setup-database` task requires these folders to create database users. If these folders do not exist, the `setup-database` task will fail.

2. Change to the directory `wp_profile_root/ConfigEngine`

3. To create the database users, type the following command:

   ```bash
   ./ConfigEngine.sh setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:

   `<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql`

**Parent topic:** Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Modifying database properties

**Next topic:** HP-UX stand-alone: Creating JCR table spaces
HP-UX stand-alone: Creating JCR table spaces

This topic provides manual instructions for creating JCR table spaces for Oracle.

- **Prerequisites**
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties
  - Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.
2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:
   - `&jcrdb` is the name of the database you created to store user data.
   - `&dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.

   Ensure that the `'` is included in the variables when you substitute the values of your environment with these variables.
   **Important:** You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

```sql
create tablespace ICMLFQ32 datafile '&dbpath./&jcrdb./data/ICMLFQ32_01.dbf' size 300M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
create tablespace ICMLNF32 datafile '&dbpath./&jcrdb./data/ICMLNF32_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
create tablespace ICMLVFO4 datafile '&dbpath./&jcrdb./data/ICMLVFO4_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
create tablespace ICMSFQ04 datafile '&dbpath./&jcrdb./data/ICMSFQ04_01.dbf' size 150M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
create tablespace ICMLSNDEX datafile '&dbpath./&jcrdb./index/ICMLSNDEX_01.dbf' size 100M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
```

A. Set the `size`, `autoextend`, and `maxsize` values according to your environment. For example, you may want to change the `maxsize` to a set value rather than `UNLIMITED`.

B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.

C. Refer to the Oracle command reference for more information about using the `create tablespace` command.
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:

- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

- Prerequisites
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties
  - Setting up databases
  - HP-UX stand-alone: Creating JCR table spaces

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.index_name.indexspace

   For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds

3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword TABLESPACE and a space. For example: community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE

Repeat this step for each domain that you are transferring.

4. Save and close dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,

   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

i5/OS: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
Configuring WebSphere Portal to use Oracle RAC

This section provides information on how to manually transfer data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- Prerequisites
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties
  - Setting up databases
  - HP-UX stand-alone: Creating JCR table spaces

Tips:
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

- The WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following: jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.

- Manually specify an RAC datasource URL in the WebSphere Application Server console. The URL should look like the following:

  jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT=1521)))(ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICE_NAME)).

- When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

Steps for transferring data to another supported database
1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.
2. Enter the following commands to validate configuration properties:
   ```bash
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ```
3. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.
4. Stop both WebSphere Application Server and the WebSphere Portal server:
   ```bash
   ./stopServer.sh server1 -username admin_userid -password admin_password
   ```
   ```bash
   ./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password
   ```
5. Transfer the database:
   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.
   A. Change to the directory `wp_profile_root/ConfigEngine`.
   B. Enter the following command:
   ```bash
   ./ConfigEngine.sh database-transfer -DWasPassword=password
   ```
   **Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:
   ```bash
   ./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
   ```
   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.
6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example:
   ```sql
   SQL> execute dbms_stats.gather_schema_stats(ownname=> 'jcr', cascade=> TRUE);
   ```
7. Specify the JDBC URL to connect to the cluster:
   A. Login to the WebSphere Application Server Administrator Console
   B. Navigate to **Resources > JDBC Providers**
   C. If there is a value in the Node field, remove it and click **Apply**.
   D. For each Oracle JDBC provider, repeat the following steps:
      1. Click the provider name.
      2. Select **Data Sources**.
      3. Click the name of the data source.
      4. Under the Oracle data source properties, modify the URL attribute according to Oracle RAC configuration:
         ```sql
         jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT =1521)) (ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICENAME)))
         ```
      5. Save your changes
8. Change to the directory `wp_profile_root/bin`.
9. Enter the following command to start the WebSphere Portal server:
   ```bash
   ./startServer.sh WebSphere_Portal
   ```

**Parent topic:** Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Assigning custom table spaces
Setting up a remote SQL Server database on Windows for a stand-alone production server

To setup a remote Microsoft SQL Server Enterprise Edition database, create user IDs and databases on a remote server. Tasks are provided to assist with creating the user IDs and the databases. Before you can use the tasks, you must modify properties files.

1. Installing SQL Server
   View the steps to install SQL Server for use with WebSphere Portal.

2. Modifying database properties
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

3. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

   The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

5. Configuring WebSphere Portal to use SQL Server 2005
   View information on manually transferring data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configuring databases
Installing SQL Server

View the steps to install SQL Server for use with WebSphere Portal.

Before you begin this task, complete the following prerequisites:

- You should have completed reviewing the Planning for SQL Server topic.
- You must install SQL Server separately from WebSphere Portal.
- WebSphere Portal requires either the DataDirect Connect for JDBC 3.7 JDBC driver or the Microsoft SQL Server JDBC Driver.
- You can obtain the DataDirect Driver from DataDirect Technologies. The JDBC driver consists of two parts: the JDBC driver libraries and JTA libraries. The JDBC driver libraries must be put on the WebSphere Portal system, while the JTA libraries are intended for SQL Server.
- You can obtain the Microsoft SQL Server JDBC Driver from Microsoft. See the Microsoft SQL Server product documentation for installation details.

The following section provides instructions for installing SQL Server for use with IBM® WebSphere® Application Server and WebSphere Portal. These steps are the same for both the DataDirect and Microsoft drivers unless noted.

1. Install SQL Server and all required patches.
2. Select the authentication mode for this installation:
   - Windows Authentication Mode
   - Mixed Mode (Windows Authentication and SQL Server Authentication)
     
     Either authentication mode allows the user who is specified as the database administrator in the wkplc_comp.properties file to log in to the SQL server. Mixed Mode authentication allows either a Windows user or a SQL server user, or both, to log in to the SQL Server.

3. In the SQL Server Setup panel, Components to Install, select the following components, which are required services for WebSphere Portal:
   - SQL Server Database Services
     - Integration Services

     The option Integration Services, creates the directory DTS/Binn, where you will need to copy files when you install Microsoft SQL Server JDBC drivers and enable XA connections.

4. Complete the installation using SQL Server documentation as a guide.
5. Enable TCP/IP connectivity in the SQL Server Configuration Manager.
6. Install the JDBC driver using one of these methods:
   - Installing DataDirect Connect for JDBC drivers on UNIX
   - Installing DataDirect Connect for JDBC drivers on Windows
   - Installing Microsoft SQL Server JDBC drivers and enabling XA connections
7. Network the connections for MSDTC in the Windows Component Services as described in Microsoft Help and Support.
8. Start SQL Server.

**Installing DataDirect Connect for JDBC drivers on UNIX**

1. Purchase and download DataDirect Connect for JDBC and save file 360connectjdbc.jar in a temporary work directory.
2. To create the required files, run the following command from the directory that contains 360connectjdbc.jar:

```
jar -xvf 360connectjdbc.jar
```

3. Run `./Installer.sh` in the same directory.

4. When installation is complete, change directory to `/installation/lib`, where `installation` is the directory in which you installed the DataDirect Connect for JDBC drivers.

5. Run the following command to change the permissions on the JAR files in the `installation/lib` directory:

```
chmod 777 *.jar
```

6. Change the ownership and group of these files by running the following commands in the `installation/lib` directory:

```
chgrp system_grp *.jar
chown root *.jar
```

Where `system_grp` is the system group as labeled by your operating system.

**Installing DataDirect Connect for JDBC drivers on Windows**

1. Purchase and download DataDirect Connect for JDBC and save file `360connectjdbc.jar` in a temporary work directory.

2. To create the required files, run the following command from the directory that contains `360connectjdbc.jar`:

```
jar -xvf 360connectjdbc.jar
```

3. Run `Installer.bat` in the same directory.

**Installing Microsoft SQL Server JDBC drivers and enabling XA connections**

1. Download and install the Microsoft SQL Server JDBC driver.

2. Copy file `sqljdbc_xa.dll` from the `xa` subdirectory to the `C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqljdbc_xa.dll` directory of the SQL Server installation.

3. Start the database server.

4. Ensure that the Distributed Transaction Coordinator has been started. The status can be verified in the list of services in the Computer Management console.

5. Start the Microsoft SQL Server Management Studio and connect to the local database engine as the system administrator, `sa`.

6. Select **File > Open > File** and select `xa_install.sql` from the subdirectory of the downloaded and extracted JDBC driver.

7. Execute the script by selecting **Query > Execute**. **Note:** Any warnings that appear in the messages section of the application window that say that stored procedures cannot be found can be safely ignored.


   a. Create a additional value in the Windows registry for WebSphere Portal by following these steps:
      - Open the Windows Registry Editor (`regedit`) and navigate to the element
        ```
        HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSDTC\XADLL
        ```
      - From the menu bar, select **Edit > New > String Value** to create a new parameter named `sqljdbc_xa.dll` in that element.
      - Change the value of the new parameter to the location of the `sqljdbc_xa.dll` file copied in Step 2 above, for example: `C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqljdbc_xa.dll`

**Enable XA Transactions In Windows Component Services**

1. From your Windows desktop, follow these steps:

2. Click **Start > Settings > Administrative Tools > Component Services.**
3. Expand the tree view to locate the computer where you want to turn on support for XA transactions (for example, My Computer).
4. Display the context menu for the computer name and click **Properties**.
5. Click **Options** and tune the Transaction Timeout that suits your environment. (The recommended minimum is 180 seconds).
6. Click **MSDTC** and click **Security Configuration**.
7. Under Security Settings, select **XA Transactions** to enable this support.
8. Click OK to save your changes.

**Note:** The installation documentation for JDBC XA connectivity refers to two known problems, see the Microsoft support site for more information:
- KB318818: Performance slows down when you use XA Transactions with SQL Server.

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server

**Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

**Prerequisites**

- Installing SQL Server

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:

- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**

The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:

- release.DbName=release
- jcr.DbName=jcrdb
- feedback.DbName=fdbkdb
- likeminds.DbName=lmdb
- community.DbName=commdb
- customization.DbName=custdb

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.

Depending on which database domain has to be configured, replace dbdomain with:

- release
- customization
- community
- jcr
- feedback
- likeminds

The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:

- dbdomain.DbType
- dbdomain.DbName
1. If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If *DbUser*, *DbUrl*, and *DbPassword* are not the same across domains, the value for *DataSourceName* must differ from the *DataSourceName* of the other domains. In other words, this value must be unique for the database domain.

2. Locate the following files and create a backup copy of each before changing any values:
   - `wp_profile_root/ConfigEngine/properties/wkplc.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_comp.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties`

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In *wkplc_comp.properties*, most properties are repeated for each domain.

3. Use a text editor to open the properties file *wkplc_comp.properties* and modify the values to correspond to your environment.
   - **A.** For *dbdomain DbType*, type *sqlserver2005*.
   - **B.** For *dbdomain DbName*, type the name of the WebSphere Portal domain database. *Note:* This value is also the database element in the *dbdomain DbUrl* property.
   - **C.** For *dbdomain DbSchema*, type the schema name of the database domain. *Note:* Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   - **D.** For *dbdomain DataSourceName*, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
     - *releaseDS*
     - *communityDS*
     - *customizationDS*
     - *jcrDS*
     - *lmdbDS*
     - *feedback*
   - **E.** For *dbdomain DbUrl*, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. *Note:* The database element of this value should match the value of *DbName*.
   - **F.** For *dbdomain DbUser*, type the user ID for the database administrator.
   - **G.** For *dbdomain DbPassword*, type the password for the database administrator.
   - **H.** For *dbdomain DBA.DbUser*, type the database administrator user ID for privileged access operations during creation of the database.
   - **I.** For *dbdomain DBA.DbPassword*, type the database administrator password for privileged access operations during creation of the database.
   - **J.** For *dbdomain DbHome*, type the root location for the database. *Note:*
     - *This value is the location to store the database files locally.*
     - *This path must use two backslashes (\) instead of a forward slash (/).*
   - **K.** For *dbdomain AdminUrl*, type the sqlserver URL without a database attached. This value is used to connect to the server for database administration operations.
   - **L.** For *dbdomain DbHostName*, type the hostname of the database.
3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For `source.domain DbType`, type of the database you are currently configured to use. The value for `source.domain DbType` is Derby by default.

B. For `source.domain DbName`, type the name of the database domain you are currently using.

C. For `source.domain DbSchema`, type current schema identifier for objects within the database for this domain.

D. For `source.domain DataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For `source.domain DbUrl`, type the url currently used to access your database.

F. For `source.domain DbUser`, type the name of the user accessing this database.

G. For `source.domain DbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

   A. For `sqlserver2005 DbDriver`, type the name of the class that SqlProcessor uses to import SQL files.

   B. For `sqlserver2005 DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For `sqlserver2005 JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

   D. For `sqlserver2005 DbConnectionPoolDataSource`, type the name of the implementation class of the connection pool data source.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

---

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server

**Previous topic:** Installing SQL Server

**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

Before you begin, ensure that the following prerequisites are met:
- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

**Prerequisites**

- Installing SQL Server
- Modifying database properties

1. Change to the directory `wp_profile_root(ConfigEngine`.
2. To create the databases, type the following command:
   ```
   ConfigEngine.bat create-database -DWasPassword=password
   ```
3. To create the database users, type the following command:
   ```
   ConfigEngine.bat setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:

   `<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql`.

**Creating users manually**

Use this alternative method for creating users if you have problems running the setup-database task that is documented for setting up a remote SQL Server 2005 database on Windows for a stand-alone production server.

Parent topic: Setting up a remote SQL Server database on Windows for a stand-alone production server

Previous topic: Modifying database properties

Next topic: HP-UX stand-alone server: Assigning custom filegroups on SQL Server 2005
Creating users manually

Use this alternative method for creating users if you have problems running the setup-database task that is documented for setting up a remote SQL Server 2005 database on Windows for a stand-alone production server.

Before you begin:

- You should have completed installing SQL Server 2005 and creating databases.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

Create the recommended database users with the SQL Server Management Studio. At least one user is required for each SQL Server 2005 instance.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Connect to your SQL Server 2005 instance.
2. Expand the tree view beneath the SQL Server instance.
3. Expand Security and right-click on Logins.
4. In the opening context menu, select New Login...
5. Enter the database user names.
7. Set a password for the selected user.
8. In the Database Access window, select the database the user must connect to at runtime. The following mappings are recommended: Table 1. List of mappings by database

<table>
<thead>
<tr>
<th>Database</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELEASE</td>
<td>RELEASEUSR</td>
</tr>
<tr>
<td>COMMUNITY</td>
<td>COMMUNITYUSR</td>
</tr>
<tr>
<td>CUSTOMIZATION</td>
<td>CUSTOMIZATIONUSR</td>
</tr>
<tr>
<td>WORKFLOW</td>
<td>WORKFLOWUSR</td>
</tr>
<tr>
<td>WMM</td>
<td>WMMDBUSR</td>
</tr>
<tr>
<td>JCRDB</td>
<td>JCR</td>
</tr>
<tr>
<td>FDBKDB</td>
<td>FEEDBACK</td>
</tr>
<tr>
<td>LMDB</td>
<td>LIKEMINDS</td>
</tr>
</tbody>
</table>

9. Click OK to save the user changes.
10. Add the role SqlJDBCXAUser to all the database users that connect to the database using XA connections. This role is available for the database "master" only. To grant users this role:
   A. Open the Microsoft SQL Server Management Studio and connect to the database instance.
   B. Expand Security > Logins underneath the database instance name.
   C. Click the user name to open the Login Properties window.
   D. Select User Mapping.
E. Select database master in the database list.
F. Select SqlJDBCXAUser.
G. Click OK.

Parent topic: Setting up databases
The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

Before you begin:
- The custom filegroups must exist prior to the execution of the database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties` file.
- For details on creating filegroups refer to the documentation for the database.

**Prerequisites**
- Installing SQL Server
- Modifying database properties
- Setting up databases

If custom filegroups are assigned, each must be assigned explicitly. The default filegroups can be used to contain database objects; however the name of the default file group must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom filegroups:

1. Determine the names of your custom filegroups.
2. Open the mapping file `wp_profile_root/PortalServer/config/tablespaces` that specifies the table space and index space for each property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.indexspace.indexspace`
   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds
3. Assign a filesystem to each entry in the mapping file. The filegroup name must be prepended by the keyword `ON` and a space. For example: `community.COMP_INST.tablespace=ON COMM8KSPACE`
   Repeat this step for each domain that you are transferring.
4. Save and close `dbdomain.space_mapping.properties`.
5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - Windows: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
   - UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server

**Previous topic:** Setting up databases

**Next topic:** Configuring WebSphere Portal to use SQL Server 2005
Configuring WebSphere Portal to use SQL Server 2005

View information on manually transferring data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- Prerequisites
  - Installing SQL Server
  - Modifying database properties
  - Setting up databases

Tips:

- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

Steps for transferring data to another supported database

1. Open a command prompt and change to the directory `wp_profile_root\ConfigEngine`.
2. Enter the following commands to validate configuration properties:
   ```
   ConfigEngine.bat validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ConfigEngine.bat validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```
3. From the same command prompt as the previous steps, change to the directory `wp_profile_root\bin`.
4. Stop both WebSphere Application Server and the WebSphere Portal server:
   ```
   Option | Description |
   |---------------|---------------|
   WebSphere Application Server | stopServer.bat server1 -username admin_userid -password admin_password |
   WebSphere Portal | stopServer.bat WebSphere_Portal -username admin_userid -password admin_password |
   ```
5. Transfer the database:
   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.
   
   A. Change to the directory `wp_profile_root\ConfigEngine`.
B. Enter the following command: `ConfigEngine.bat database-transfer -DWasPassword=password`

**Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command: `ConfigEngine.bat database-transfer -DTransferDomainList=jcr -DWasPassword=password`

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. Change to the directory `wp_profile_root\bin`.

7. Enter the following command to start the WebSphere Portal server: `startServer.bat WebSphere_Portal`

8. Update the SQL Server 2005 statistics for Portal, and JCR databases by opening SQL Server Management Studio, selecting **New Query**, and running the following query: `use db_name exec sp_updatestats @resample='resample';`

---

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server

**Previous topic:** HP-UX stand-alone server: Assigning custom filegroups on SQL Server 2005
Verifying databases

After you configure IBM® WebSphere® Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.

- Prerequisites
  - [Technotes for database connectivity issues](#)

You can verify the database connection using IBM WebSphere Application Server or by opening WebSphere Portal in a browser.

- To verify that the WebSphere Portal application server is running by using WebSphere Application Server, complete these steps:

  1. Open the WebSphere Application Server administrative console by entering the following address in a browser:

     http://hostname.example.com:10027/ibm/console

     where hostname.example.com is the fully qualified host name of the machine where WebSphere Portal is running and 10027 is the default transport port that is created by WebSphere Application Server.

  2. Log into the administrative console.

  3. Depending on your version of WebSphere Application Server, click the appropriate option:

     - For WebSphere Application Server Version 6.1: Click Resources & JDBC Providers.
     - For WebSphere Application Server Version 7.0: Click Resources > JDBC > JDBC Providers

  4. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.

  5. Select the name of the data source that is defined in wkplc_comp.properties. The default data source is wpdbDS.

  6. Select the name of the JDBC provider that is specified in wkplc_dbtype.properties. The default JDBC provider is wpdbJDBC_dbtype, where dbtype is replaced by the value that matches your environment.

  7. Click Test Connection to verify the database connection. If configuration parameters have been changed, you might need to restart WebSphere Application Server for the test to complete.

- To verify that the WebSphere Portal application server is running by opening WebSphere Portal in a browser, enter the following URL in a supported browser:http://hostname.example.com:10040/wps/portal

  where hostname.example.com is the fully qualified host name of the machine where WebSphere Portal is running and 10040 is the default transport port that is created by WebSphere Application Server.

Parent topic: Configuring databases
Preparing a remote Web server when portal is installed on HP-UX

Install and configure the Web server plug-in, provided by IBM® WebSphere® Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

- **Prerequisites**
  - Preparing your HP-UX operating system
  - Installing WebSphere Portal on HP-UX
  - Configuring databases

Perform the following steps to install and configure your Web server:

1. Install and configure the Web server; refer to the Web server documentation for information.
2. If using IBM Lotus® Domino®, edit the NOTES.INI file on the Web server. Set the HTTPEnableConnectorHeaders parameter to 1.
3. If using IBM HTTP Server or Apache Server, edit the httpd.conf file on the Web server. Set the AllowEncodedSlashes directive to on; the directive should be added at the root level as a global directive.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>See the appropriate HTTP Server documentation</td>
<td>IBM HTTP Server</td>
</tr>
<tr>
<td>See the appropriate Apache Server documentation</td>
<td>AllowEncodedSlashes directives</td>
</tr>
</tbody>
</table>

4. Stop the Web server.

5. Install and configure the Web server plug-in on the system where the Web server is located using the Plug-ins installation wizard provided with WebSphere Application Server. Refer to Selecting a Web server topology diagram and roadmap for information. Note; The WebSphere Application Server Web server instructions are for the default profile. If not using the default profile, refer to Configuring a Web server for a non-default profile.

   **If using WebDAV:** After successfully installing the Web server plug-in, locate and open your plugin-cfg.xml file and set AcceptAllContent to true.

   **Important:** Depending on how you use the Web server, you may need to adjust the ServerIOTimeout value, which defines how long the plug-in should wait for a response from the application. The recommended minimum value is 60 but you may need to adjust this value higher if you are retrieving data from a database. To update this value, locate and open your plugin-cfg.xml file and set ServerIOTimeout to a value that is appropriate for your business needs. For additional information, see Common questions about the Web server plug-in.

6. If you are using Sun Web Server Version 7 update 8, read Technote 1448262 and perform the steps to resolve the HTTP 408/409 error.

7. Start the Web server.

**Parent topic:** Setting up a stand-alone server on HP-UX  
**Previous topic:** Configuring databases  
**Next topic:** Configuring WebSphere Portal to use a user registry on HP-UX
Configuring WebSphere Portal to use a user registry on HP-UX

Configure user registry security on IBM® WebSphere® Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

Prior to configuring security, you should use the IBM WebSphere Application Server backupConfig task to create and store a backup of the IBM WebSphere Portal configuration; see backupConfig command for information.

- Prerequisites
  - Preparing your HP-UX operating system
  - Installing WebSphere Portal on HP-UX
  - Configuring databases

Perform the following tasks to configure WebSphere Portal to use a user registry:

1. Preparing user registries on HP-UX
   Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

2. Choosing your user registry model on HP-UX
   Choose between securing IBM WebSphere Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

3. Adapting the attribute configuration
   After installing IBM WebSphere Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

4. Stand-alone server: Configuring WebSphere Portal to use dynamic groups on HP-UX
   By default, WebSphere Portal is enabled for static groups. However, the Virtual Member Manager (VMM) allows users to be members of either static or dynamic groups. Static groups are those where a persistent binding exists between a group and its members. Dynamic groups are those where a search query is defined to retrieve the members of a group. If you have your LDAP server configured to use dynamic groups, complete the steps in this task for WebSphere Portal to use dynamic group queries when you setup your LDAP server.

5. Stand-alone server: Enabling referrals for your LDAP user registry on HP-UX
   Referrals redirect object requests from one LDAP server to another when objects do not exist or cannot be located in a particular directory tree. You should enable referrals if your environment has more than one user registry existing on multiple servers or domains.

Parent topic: Setting up a stand-alone server on HP-UX

Previous topic: Preparing a remote Web server when portal is installed on HP-UX

Next topic: Tune your servers

Related tasks
Managing your user registry on an HP-UX
Preparing user registries on HP-UX

Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

Choose the appropriate LDAP server to install and setup:

- **Preparing a Domino Directory server**
  If you plan to use a Domino Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a SecureWay Security Server**
  If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Novell eDirectory**
  If you plan to use a Novell eDirectory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Sun Java System Directory Server**
  If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Tivoli Directory Server**
  If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

**Parent topic:** Configuring WebSphere Portal to use a user registry on HP-UX

**Next topic:** Choosing your user registry model on HP-UX
Preparing a Domino Directory server

If you plan to use a Domino Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Domino Directory:

1. Perform the following steps to install the Domino Directory:
   A. Go to Lotus Domino documentation.
   B. Select the appropriate version tab for your product.
   C. Click the Download/View online link for the Lotus Domino for multiple platforms Information Center.
   D. Click Domino Administrator Help > Installation > Installing and setting up Domino servers > Server installation > Installing Domino on Unix and perform this task.
   E. Click Domino Administrator Help > Installation > Installing and setting up Domino servers > The Domino server setup program and perform this task.

2. Perform the following steps as a guide to create the WebSphere Portal administrative user:
   A. Navigate to the People view of the Domino Directory and then click Add Person.
   B. Enter the following values in the New Person form to create the wpsbind user:
      - Last Name
      - wpsbind
      - User Name
      - wpsbind/DominoDomain, where DominoDomain is your Lotus Domino Internet domain
      - wpsbind
      Note: Make sure you enter two values in the User Name field, where the first value includes the Lotus Domino domain.
      - Short name/UserID
      - wpsbind
      - Internet password
      - wpsbind
   C. Click Save and Close to save the new person record for wpsbind and return to the People view.
   D. Click Add Person and enter the following values in the New Person form to create the wpsadmin user:
      - Last Name
      - wpsadmin, where wpsadmin in the user ID for the WebSphere Portal Administrator
      - User Name
      - wpsadmin/DominoDomain, where DominoDomain is your Lotus Domino Internet domain
      - wpsadmin
      Note: Make sure you enter two values in the User Name field, where the first value includes the Lotus Domino domain.
      - Short name/UserID
      - wpsadmin
      - Internet password
      - wpsadmin
   E. Click Save and Close to save the new person record for wpsadmin and return to the People view.
   F. Navigate to the Groups view and click Add Group.
   G. Enter the following values in the New Group form on the Basic tab:
      - Group name
- *wpsadmins*  
  **Note:** If you plan to configure WebSphere Portal for multiple user registries and your Lotus Domino LDAP will share a realm with another user registry, you must use the hierarchical naming convention for the group names, for example: wpsadmins/DominoDomain, to avoid unexpected results during WebSphere Portal runtime.

- **Group type**  
  - Multi-purpose

- **Members**  
  - wpsbind/DominoDomain
  - wpsadmin/DominoDomain

  **Note:** You can add additional administrator users if required.

H. Click **Save and Close** to save the wpsadmins group with the wpsbind and wpsadmin users as members.

3. Perform the following steps to update the access control list for the Domino Directory:
   A. Open the names.nsf file in the Lotus Domino Administrator or Lotus Notes client.
   B. Click **File > Application > Access Control** from the main menu to open the access control list for the file.
   C. In the **Access Control List > Basics** panel, ensure that the wpsadmins group has either Author or Editor access.
   D. Add the following Role Types to the wpsadmins group:
      - GroupCreator
      - GroupModifier
      - UserCreator
      - UserModifier
   E. Click **OK**.

**Parent topic:** Preparing user registries on HP-UX
Preparing a SecureWay Security Server

If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Sun Java System Directory Server:

1. Install Sun Java System Directory Server; refer to Oracle software for information.

2. Perform the following steps to create the WebSphere Portal administrative user:

   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.

   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.

   C. Replace every dc=yourco,dc=com with your suffix.

   D. Replace any prefixes and suffixes that are unique to your LDAP server.

   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.

   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.

   G. Save your changes.

   H. Follow the instructions provided with your directory server to import the LDIF file.

   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on HP-UX
Preparing a Novell eDirectory

If you plan to use a Novell eDirectory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Novell eDirectory:

1. Install Novell eDirectory; refer to eDirectory for information.
2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every dc=yourco,dc=com with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupName.
   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.
Preparing a Sun Java System Directory Server

If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Sun Java System Directory Server:

1. Install Sun Java System Directory Server; refer to Oracle software for information.
2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every dc=yourco,dc=com with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.
   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on HP-UX
Preparing a Tivoli Directory Server

If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Tivoli Directory Server:

   
   **Note:** Due to a restriction in Tivoli Directory Server, users or groups must not contain a Turkish uppercase dotted I or lowercase dotted i in the DN as this will prevent correct retrieval of that user or group.

2. Perform the following steps to create the WebSphere Portal administrative user:
   
   A. Optional: Perform the following steps to create a new directory suffix:
      
      1. Click the **Server Administration** folder, located in the left-hand navigation of the directory server console.
      2. Click the **Manage Server Properties** folder under the Server Administration folder and then select **Suffixes** on the main page.
      3. Type the **Base DN** name for the suffix; for example: *dc=yourcompany,dc=com*.
      4. Click **Add**.
      5. Click **OK** to save your changes.
      6. Stop and restart the LDAP server.

   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the *PortalUsers.ldif* file as a working example and adapted appropriately to work with your LDAP server.
      - Use the *ContentUsers.ldif* file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.

   C. Replace every *dc=yourco,dc=com* with your suffix.

   D. Replace any prefixes and suffixes that are unique to your LDAP server.

   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.

   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the **objectclasses** to **accessGroup**. If using Tivoli Access Manager Version 6, set the **objectclasses** to **groupName**.

   G. Save your changes.

   H. Follow the instructions provided with your directory server to import the LDIF file.

   I. Stop and restart the LDAP server.

**Parent topic:** Preparing user registries on HP-UX
Choosing your user registry model on HP-UX

Choose between securing IBM® WebSphere® Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

- **Prerequisites**
  - Preparing user registries on HP-UX

Choose one of the following user registry models:

- **Configuring a stand-alone LDAP user registry on HP-UX**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Configuring the default federated repository on HP-UX**
  Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

**Parent topic:** Configuring WebSphere Portal to use a user registry on HP-UX

**Previous topic:** Preparing user registries on HP-UX

**Next topic:** Adapting the attribute configuration
Configuring a stand-alone LDAP user registry on HP-UX

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Configuring a stand-alone LDAP user registry on HP-UX**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry to store all user account information for authorization.

- **Configuring a stand-alone LDAP user registry over SSL on HP-UX**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

**Parent topic:** Choosing your user registry model on HP-UX
Configuring a stand-alone LDAP user registry on HP-UX

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry to store all user account information for authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps. If you need to rerun the wp-modify-ldap-security task to change the LDAP repositories or because the task failed, you must choose a new name for the realm using the standalone.ldap.realm parameter or you can set ignoreDuplicateIds=true in the wkpc.properties file, before rerunning the task.

Perform the following steps to configure a standalone LDAP user registry:

**Note:** Use the wp_security_xxx.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_security_xxx.properties helper file.

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

2. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM Standalone LDAP configuration heading:

**Note:** See the properties file for specific information about the required parameters and for advanced parameters.

- standalone.ldap.id
- standalone.ldap.host
- standalone.ldap.port
- standalone.ldap.blindDN
- standalone.ldap.bindPassword
- standalone.ldap.idServerType
- standalone.ldap.userIdMap
- standalone.ldap.groupIdMap
- standalone.ldap.groupMemberIdMap
- standalone.ldap.userFilter
- standalone.ldap.groupFilter
- standalone.ldap.serverId
- standalone.ldap.serverPassword
- standalone.ldap.realm
- standalone.ldap.primaryAdminId
- standalone.ldap.primaryAdminPassword
- standalone.ldap.primaryPortalAdminId
- standalone.ldap.primaryPortalAdminPassword
- standalone.ldap.primaryPortalAdminGroup
- standalone.ldap.baseDN
3. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading:

   - `standalone.ldap.et.group.objectClasses`
   - `standalone.ldap.et.group.objectClassesForCreate`
   - `standalone.ldap.et.group.searchBases`
   - `standalone.ldap.et.personaccount.objectClasses`
   - `standalone.ldap.et.personaccount.objectClassesForCreate`
   - `standalone.ldap.et.personaccount.searchBases`

   **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

4. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attributes heading:

   - `standalone.ldap.gm.groupMemberName`
   - `standalone.ldap.gm.objectClass`
   - `standalone.ldap.gm.scope`
   - `standalone.ldap.gm.dummyMember`

   **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

5. Required: Enter a value for the following required RDN® parameters in the `wkplc.properties` file under the Default parent, RDN attribute heading:

   - `standalone.ldap.personAccountParent`
   - `standalone.ldap.groupParent`
   - `standalone.ldap.personAccountRdnProperties`
   - `standalone.ldap.groupRdnProperties`

   **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

6. Save your changes to the `wkplc.properties` file.

7. Optional: Enter the following Web content authors parameters in the `wkplc_comp.properties` file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product:

   - `WcmContentAuthorsGroupId`
   - `WcmContentAuthorsGroupCN`

   **Note:** See the `wkplc_comp.properties` file for specific information about the required parameters and for advanced parameters.

8. Run the `./ConfigEngine.sh validate-standalone-ldap -DWasPassword=password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, `WasPassword` is the value entered during installation and not a value found in your LDAP user registry.

   **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now? Press y then Enter.

9. Run the `./ConfigEngine.sh wp-modify-ldap-security -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to set the stand-alone LDAP user registry.

10. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

11. Run the `./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

12. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the
LDAP directory.

Note: This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

A. Edit the `wp_profile_root`/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties file.

B. Add the following lines to the file:

```
uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
```

Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `./ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory. Note: Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

13. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

Parent topic: Configuring a stand-alone LDAP user registry on HP-UX

Related tasks

- Adapting the attribute configuration
- Using the member fixer tool with IBM Lotus Web Content Management
- Starting and stopping servers, deployment managers, and node agents
Configuring a stand-alone LDAP user registry over SSL on HP-UX

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to configure a standalone LDAP user registry over SSL. **Note:** Use the wp_security_xxx.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_security_xxx.properties helper file.

1. To specify the LDAP server’s SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:

   - **Server trust store**
     - A. Add the certificate to the trust store:
       1. Log in to the WebSphere Application Server Administrative Console.
       2. Navigate to Security > SSL certificate and key management > SSL configurations.
       3. Click the appropriate SSL configuration from the list. For example,
          - Stand-alone environments: NodeDefaultSSLSettings
          - Clustered environments: CellDefaultSSLSettings

          **Clustered environments:** Ensure the setting for SSL configuration for outbound connection matches your SSL settings.

        4. Click Key stores and certificates.
        5. Click the appropriate trust store from the list; for example, NodeDefaultTrustStore.
        6. Click Signer certificates, click Add, and then enter the following information:
           - Type the Alias the key store uses for the signer certificate.
           - Type the File name where the signer certificate is located.
        7. Click OK and then click Save to save the changes to the master configuration.

   B. Retrieve the certificate from the port:
      1. Log in to the WebSphere Application Server Administrative Console.
      2. Navigate to Security > SSL certificate and key management > SSL configurations.
      3. Click the appropriate SSL configuration from the list. For example,
         - Stand-alone environments: NodeDefaultSSLSettings
Clustered environments: 

**CellDefaultSSLSettings**

Clustered environments: Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.

4. Click **Key stores and certificates**.

5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.

6. Click **Signer certificates**, click **Retrieve from port**, and then enter the following information:
   - Type the **Host** name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the **Port** used when attempting to retrieve the signer certificate.
   - Type the **Alias** the key store uses for the signer certificate.

7. Click **Retrieve signer information** to retrieve the certificate from the port.

8. Click **OK** and then click **Save** to save the changes to the master configuration.

**Client trust store**:

This task might report an error, but it does successfully update the trust store. You can ignore the error message.

A. See Secure installation for client signer retrieval.

B. Run the retrieveSigners task from the **wp_profile_root/bin** directory; see retrieveSigners command for information. In a deployed environment, you will need to run the retrieveSigners task, for any federated node, against the Deployment Manager. **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

C. Update the trust store properties file:

1. Use a text editor to open the **ssl.client.props** file, located in the **wp_profile_root/properties** directory.

2. Change the **com.ibm.ssl.trustStore** parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter
   
   ```
   com.ibm.ssl.trustStore=${CONFIG_ROOT}\cells\wpsbvt\nodes\wpsbvt\trust.p12
   ```
   
   to use the default trust store.

3. Save your changes.

2. Use a text editor to open the **wkplc.properties** file, located in the **wp_profile_root/ConfigEngine/properties** directory.

3. Required: Enter a value for the following required parameters in the **wkplc.properties** file under the VMM Standalone LDAP configuration heading. **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - **standalone.idap.id**
   - **standalone.idap.host**
   - **standalone.idap.port**
   - **standalone.idap.blnDN**
   - **standalone.idap.bindPassword**
   - **standalone.idap.idapServerType**
   - **standalone.idap.userldMap**
   - **standalone.idap.groupldMap**
   - **standalone.idap.groupldMap**
   - **standalone.idap.userFilter**
   - **standalone.idap.groupFilter**
   - **standalone.idap.serverld**
   - **standalone.idap.serverPassword**
   - **standalone.idap.realm**
   - **standalone.idap.primaryAdminId**
   - **standalone.idap.primaryAdminPassword**
   - **standalone.idap.primaryPortalAdminId**
- standalone.ldap.primaryPortalAdminPassword
- standalone.ldap.primaryPortalAdminGroup
- standalone.ldap.baseDN

4. Required: Enter a value for the following required entity types parameters in the wkplc.properties file under the LDAP entity types heading.:Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.et.group.objectClasses
   - standalone.ldap.et.group.objectClassesForCreate
   - standalone.ldap.et.group.searchBases
   - standalone.ldap.et.personaccount.objectClasses
   - standalone.ldap.et.personaccount.objectClassesForCreate
   - standalone.ldap.et.personaccount.searchBases

5. Required: Enter a value for the following required group member parameters in the wkplc.properties file under the Group member attributes heading.:Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.gm.groupMemberName
   - standalone.ldap.gm.objectClass
   - standalone.ldap.gm.scope
   - standalone.ldap.gm.dummyMember

6. Required: Enter a value for the following required relative distinguished name (RDN®) parameters in the wkplc.properties file under the Default parent, RDN attribute heading.:Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.personAccountParent
   - standalone.ldap.groupParent
   - standalone.ldap.personAccountRdnProperties
   - standalone.ldap.groupRdnProperties

7. Enter a value for the following parameters to enable Secure Socket Layers (SSL):.Note: See the properties file for specific information about the required parameters and for advanced parameters.
   Required parameters:
   - standalone.ldap.sslEnabled
   - standalone.ldap.sslConfiguration
   Optional parameters:
   - standalone.ldap.certificateMapMode
   - standalone.ldap.certificateFilter

8. Save your changes to the wkplc.properties file.

9. Run the ./ConfigEngine.sh validate-standalone-ldap -DWasPassword=password task to validate your LDAP server settings.:Attention: If you have not deleted the default file repository, WasPassword is the value entered during installation and not a value found in your LDAP user registry.
   :Note: During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press y then Enter.

10. Run the ./ConfigEngine.sh wp-modify-ldap-security -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to set the stand-alone LDAP user registry.

11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

12. Run the ./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to check that all defined attributes are available in the configured
LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

13. Configure your Web server over SSL. Navigate to **Configuring WebSphere Portal > Additional security features > Configuring SSL > Setting up SSL** for information.

14. Log on to the WebSphere Application Server Administrative Console and navigate to **Security > SSL certificate and key management**. Click the **Use the United States Federal Information Processing Standard (FIPS) algorithms.** check box to enable FIPS.

15. Enable TLS in your internet browser, located under **Tools > Options > Advanced.**

**Parent topic:** Configuring a stand-alone LDAP user registry on HP-UX

**Related tasks**

Adapting the attribute configuration

Starting and stopping servers, deployment managers, and node agents
Configuring the default federated repository on HP-UX

Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

Perform the following tasks to add user registries to the default federated repository:

- **Configuring a federated LDAP user registry on HP-UX**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Adding a database user registry on HP-UX**
  Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

- **Adding realm support on HP-UX**
  A realm is a group of users from one or more user registries that form a coherent group within IBM WebSphere Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

Parent topic: Choosing your user registry model on HP-UX
Configuring a federated LDAP user registry on HP-UX

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Adding an LDAP user registry on HP-UX**
  Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time. If IBM Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is \texttt{cn=groupName} and the hierarchical format is \texttt{cn=groupName,o=root}. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as \texttt{wpsadmin}, this ID cannot exist in the LDAP you are adding.

- **Adding an LDAP user registry over SSL on HP-UX**
  Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

**Parent topic:** Configuring the default federated repository on HP-UX
Adding an LDAP user registry on HP-UX

Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

If IBM® Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=groupName` and the hierarchical format is `cn=groupName, o=root`. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as `wpsadmin`, this ID cannot exist in the LDAP you are adding.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

**Note:** Use the `wp_add_federated_xxx.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_add_federated_xxx.properties` helper file.

1. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
2. Required: Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated LDAP Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `federated.ldap.id`
   - `federated.ldap.host`
   - `federated.ldap.port`
   - `federated.ldap.bindDN`
   - `federated.ldap.bindPassword`
   - `federated.ldap.ldapServerType`
   - `federated.ldap.baseDN`
3. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `federated.ldap.et.group.objectClasses`
   - `federated.ldap.et.group.objectClassesForCreate`
   - `federated.ldap.et.group.searchBases`
   - `federated.ldap.et.personaccount.objectClasses`
   - `federated.ldap.et.personaccount.objectClassesForCreate`
   - `federated.ldap.et.personaccount.searchBases`
4. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attribute heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `federated.ldap.gm.groupMemberName`
   - `federated.ldap.gm.objectClass`
   - `federated.ldap.gm.scope`
   - `federated.ldap.gm.dummyMember`

5. Save your changes to the `wkplc.properties` file.

6. Optional: Enter the following Web content authors parameters in the `wkplc_comp.properties` file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. **Note:** See the `wkplc_comp.properties` file for specific information about the required parameters and for advanced parameters.
   - `WcmContentAuthorsGroupId`
   - `WcmContentAuthorsGroupCN`

7. Run the `.ConfigEngine.sh validate-federated-ldap` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, `WasPassword` is the value entered during installation and not a value found in your LDAP user registry. **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press y then Enter.

8. Run the `.ConfigEngine.sh wp-create-ldap` task, from the `wp_profile_root/ConfigEngine` directory, to add an LDAP user registry to the default federated repository. **Note:** Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

10. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:
    A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
    B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
       - `id`
       - `baseDN`
       - `nameInRepository`
    C. Save your changes to the `wkplc.properties` file.
    D. Run the `.ConfigEngine.sh wp-create-base-entry` task, from the `wp_profile_root/ConfigEngine` directory, to create a base entry in a repository.
    E. Stop and restart all necessary servers to propagate your changes.

11. Optional: Run the `.ConfigEngine.sh wp-query-repository` task, from the `wp_profile_root/ConfigEngine` directory, to list the names and types of configured repositories.

12. Run the `.ConfigEngine.sh wp-validate-federated-ldap-attribute-config` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between
WebSphere Portal and the LDAP server.

13. Perform the following steps to update the user registry where new users and groups are stored:

   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
      - `personAccountParent`
      - `groupParent`
      - `personAccountRdnProperties`
      - `groupRdnProperties`

      The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:
      - `personAccountParent=dc=yourco,dc=com`
      - `groupParent=dc=yourco,dc=com`

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to delete the old attributes before adding the new attributes.

   E. Stop and restart all necessary servers to propagate your changes.

14. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for `realmName` or leave blank to update the default realm.

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login.**Note:** After running this task to enable the full distinguished name login, you can run the `./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

   E. Stop and restart all necessary servers to propagate your changes.

15. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

   **Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

   A. Edit the `wp_profile_root/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties` file.

   B. Add the following lines to the file:

   ```
   uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
   cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
   ```

   Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.
C. Save your changes and close the file.

D. Run the `./ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: *Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management*

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.idap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

16. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

17. Optional: **This step is required in a production environment.** Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

   **Important:**
   - Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
   - Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

   **Note:** If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

   A. Run the following task from the `wp_profile_root/ConfigEngine` directory: `./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword`. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` parameter.

   B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

   C. Run the following task from the `wp_profile_root/ConfigEngine` directory: `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid`. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

   D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

18. Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it.

   The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by
duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: *Deleting the repository.*

**Parent topic:** Configuring a federated LDAP user registry on HP-UX

**Related tasks**
- Adapting the attribute configuration
- Deleting the repository on HP-UX
- Starting and stopping servers, deployment managers, and node agents

**Related information**
- User IDs and passwords
Adding an LDAP user registry over SSL on HP-UX

Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry over SSL to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

**Note:** Use the `wp_add_federated_xxx.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_add_federated_xxx.properties` helper file.

1. To specify the LDAP server's SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:
   - **Server trust store**
     - A. Add the certificate to the trust store:
       1. Log in to the WebSphere Application Server Administrative Console.
       2. Navigate to Security > SSL certificate and key management > SSL configurations.
       3. Click the appropriate SSL configuration from the list. For example,
          - Stand-alone environments: **NodeDefaultSSLSettings**
          - Clustered environments: **CellDefaultSSLSettings**
          **Clustered environments:** Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.
       4. Click **Key stores and certificates**.
       5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.
       6. Click **Signer certificates**, click **Add**, and then enter the following information:
          - Type the **Alias** the key store uses for the signer certificate.
          - Type the **File name** where the signer certificate is located.
       7. Click **OK** and then click **Save** to save the changes to the master configuration.
   - **B. Retrieve the certificate from the port:**
     1. Log in to the WebSphere Application Server Administrative Console.
     2. Navigate to Security > SSL certificate and key management > SSL configurations.
     3. Click the appropriate SSL configuration from the list. For example,
        - Stand-alone environments: **NodeDefaultSSLSettings**

Clustered environments: **CellDefaultSSLSettings**

**Clustered environments:** Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.

4. Click **Key stores and certificates**.

5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.

6. Click **Signer certificates**, click **Retrieve from port**, and then enter the following information:
   - Type the **Host** name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the **SSL Port** used when attempting to retrieve the signer certificate.
   - Type the **Alias** the key store uses for the signer certificate.

7. Click **Retrieve signer information** to retrieve the certificate from the port.

8. Click **OK** and then click **Save** to save the changes to the master configuration.

**Client trust store**

- **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

  A. See Secure installation for client signer retrieval.

  B. Run the **retrieveSigners** task from the **wp_profile_root/bin** directory; see **retrieveSigners command** for information. In a deployed environment, you will need to run the **retrieveSigners** task, for any federated node, against the Deployment Manager. **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

C. Update the trust store properties file:

   1. Use a text editor to open the **ssl.client.props** file, located in the **wp_profile_root/properties** directory.

   2. Change the **com.ibm.ssl.trustStore** parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter `com.ibm.ssl.trustStore=${CONFIG_ROOT}/cells/wpsbvt/nodes/wpsbvt/trust.p12` to use the default trust store.

   3. Save your changes.

2. Use a text editor to open the **wkplc.properties** file, located in the **wp_profile_root/ConfigEngine/properties** directory.

3. Required: Enter a value for the following required parameters in the **wkplc.properties** file under the VMM Federated LDAP Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - **federated.ldap.id**
   - **federated.ldap.host**
   - **federated.ldap.port**
   - **federated.ldap.bindDN**
   - **federated.ldap.bindPassword**
   - **federated.ldap.IdapServerType**
   - **federated.ldap.baseDN**

4. Required: Enter a value for the following required entity types parameters in the **wkplc.properties** file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - **federated.ldap.et.group.objectClasses**
   - **federated.ldap.et.group.objectClassesForCreate**
   - **federated.ldap.et.group.searchBases**
   - **federated.ldap.et.personaccount.objectClasses**
   - **federated.ldap.et.personaccount.objectClassesForCreate**
   - **federated.ldap.et.personaccount.searchBases**
5. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attribute heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `federated.idap.gm.groupMemberName`
   - `federated.idap.gm.objectClass`
   - `federated.idap.gm.scope`
   - `federated.idap.gm.dummyMember`

6. Enter a value for the following parameters to enable Secure Socket Layers (SSL): **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   **Required parameters:**
   - `federated.idap.sslEnabled`
   - `federated.idap.sslConfiguration`

   **Optional parameters:**
   - `federated.idap.certificateMapMode`
   - `federated.idap.certificateFilter`

7. **Save your changes to the `wkplc.properties` file.**

8. Optional: Enter the following Web content authors parameters in the `wkplc_comp.properties` file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. **Note:** See the `wkplc_comp.properties` file for specific information about the required parameters and for advanced parameters.
   - `WcmContentAuthorsGroupId`
   - `WcmContentAuthorsGroupCN`

9. Run the `./ConfigEngine.sh validate-federated-ldap -DWasPassword=password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, **WasPassword** is the value entered during installation and not a value found in your LDAP user registry.

   **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press `y` then `Enter`.

10. Run the `./ConfigEngine.sh wp-create-ldap -DWasPassword=password` task, from the `wp_profile_root` directory, to add an LDAP user registry to the default federated repository. **Note:** Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents.*

12. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:
   
   **A.** Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   **B.** Enter a value for the following required parameters in the `wkplc.properties` file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms:

   **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `id`
   - `baseDN`
- nameInRepository
C. Save your changes to the wkplc.properties file.
D. Run the ./ConfigEngine.sh wp-create-base-entry -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to create a base entry in a repository.
E. Stop and restart all necessary servers to propagate your changes.
13. Optional: Run the ./ConfigEngine.sh wp-query-repository -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to list the names and types of configured repositories.
14. Run the ./ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

15. Perform the following steps to update the user registry where new users and groups are stored:

**Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

**Attention:** During installation, the default file repository creates a default value in the personAccountRdnProperties and groupRdnProperties parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
B. Enter a value for the following required parameters in the wkplc.properties file under the VMM supported entity types configuration heading:

**Note:** See the properties file for specific information about the required parameters and for advanced parameters.
- personAccountParent
- groupParent
- personAccountRdnProperties
- groupRdnProperties

The parameters groupParent and personAccountParent must be set to the same value. For example:
- personAccountParent=dc=yourco,dc=com
- groupParent=dc=yourco,dc=com

C. Save your changes to the wkplc.properties file.
D. Run the ./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to delete the old attributes before adding the new attributes.
E. Stop and restart all necessary servers to propagate your changes.
16. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
B. Enter a value for realmName or leave blank to update the default realm.
C. Save your changes to the wkplc.properties file.
D. Run the ./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password task, located in the wp_profile_root/ConfigEngine directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the ./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password task to disable the feature.
E. Stop and restart all necessary servers to propagate your changes.
17. Configure your Web server over SSL. Navigate to Configuring WebSphere Portal > Additional security features > Configuring SSL > Setting up SSL for information.
18. Log on to the WebSphere Application Server Administrative Console and navigate to Security > SSL certificate and key management. Click the Use the United States Federal Information Processing Standard (FIPS) algorithms. check box to enable FIPS.

19. Enable TLS in your internet browser, located under Tools > Options > Advanced.

20. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

   Note: This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the configure-express task.

   A. Edit the wp_profile_root

   /PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties file.

   B. Add the following lines to the file:

   guid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
   cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN

   Replace portal_admin_DN with the distinguished name of the portal administrator and content_authors_group_DN with the distinguished name of the content authors group used during LDAP configuration.

   C. Save your changes and close the file.

   D. Run the ./ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm= -DPortalAdminPwd= -DWasPassword= task, located in the wp_profile_root/ConfigEngine directory. Note: Choose the appropriate value to enter for realm_name depending on the type of LDAP user registry you configured. Table 1. Value for realm_name when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for realm_name should match the value for standalone.ldap.realm in the wkplc.properties file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for realm_name should match the value for federated.realm in the wkplc.properties file. If the value for federated.realm is empty, use defaultWIMFileBasedRealm as the default value.</td>
</tr>
</tbody>
</table>

21. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

22. Optional: This step is required in a production environment. Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

   Important:

   - Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
   - Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

   Note: If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.
A. Run the following task from the `wp_profile_root/ConfigEngine` directory:

```
./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword
```

**Important:** You must provide the full distinguished name (DN) for the `newAdminId` parameter.

B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

C. Run the following task from the `wp_profile_root/ConfigEngine` directory:

```
./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid
```

**Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

23. Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it. The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: `Deleting the repository`.

**Parent topic:** Configuring a federated LDAP user registry on HP-UX

**Related tasks**
- Adapting the attribute configuration
- Deleting the repository on HP-UX
- Starting and stopping servers, deployment managers, and node agents

**Related information**
- User IDs and passwords
Adding a database user registry on HP-UX

Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add a database user registry to the default federated repository; you must repeat these steps for each additional database user registry that you plan to add. **Note:** Use the `wp_add_DB.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_add_DB.properties` helper file.

1. Set up a new database, including creating a new user with appropriate database privileges for accessing the database:

   **Instructions for setting up databases:** Refer to the appropriate documentation for the type of database you want to set up.

   **Consulting your database administrator:** The task of setting up a new database is typically performed by a database administrator. However, the following steps are provided for your reference in the event you are creating a stand-alone database for testing or demonstration purposes. Consult your database administrator before proceeding with the following steps if you plan to create a database for a production environment.

   **Table 1. Steps for creating a new database to use as a database user registry.**

<table>
<thead>
<tr>
<th>Database</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td>Perform the following steps to create a DB2 database: Install DB2. Enter the following database tuning commands:</td>
</tr>
</tbody>
</table>
2. Perform the following steps to define the **DbDriver** and **DbLibrary** parameter values:

   A. Navigate to the following directory: `wp_profile_root/ConfigEngine/properties`
   
   B. Locate and open `wkplc_dbtype.properties` with any text editor.
   
   C. Enter a value for the following parameters under the appropriate database type properties heading:
      - `db_type_DbDriver`
      - `db_type_DbLibrary`
   
   D. Save your changes.

3. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

4. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated Database Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - `federated.db.DataSourceName`
   - `federated.dbDbType`
   - `federated.db.DbUrl`
   - `federated.db.Id`
   - `federated.db.baseDN`
   - `federated.db.DbUser`
   - `federated.db.DbPassword`
   - `federated.db.DbName`

5. Change the value for the **com.ibm.SOAP.requestTimeout** parameter to 1000.

   A. Navigate to the following directory: `wp_profile_root/properties`
   
   B. Locate and open `soap.client.props` with any text editor.
   
   C. Locate the `com.ibm.SOAP.requestTimeout` parameter and change the value to 1000.
   
   D. Save and close `soap.client.props`.

6. Perform the following steps in a clustered environment:

   A. Run the `./ConfigEngine.sh wp-prep-vmm-db-secured-environment -DWasPassword=password -DDbDomain=federated.db -Ddb_type.DmgrDbLibrary=local path of the database jars on the Deployment`
Manager
- DDmgrNodeName=dmgr_node_name task from the wp_profile_root/ConfigEngine directory to create the local Deployment Manager WebSphere variable used to access the database jars. Note: The db_type in db_type .DmgrDbLibrary should be set to the type of database you are using, for example db2. The local full path of the database jars on the Deployment Manager should be one of the following options:

- DB2 Type 2 driver: db2java.zip
- DB2 Type 4 driver: db2jcc.jar; db2jcc_license_cu.jar
- DB2 for z/OS Type 2 driver: db2java.zip
- DB2 for z/OS Type 4 driver: db2jcc.jar; db2jcc_license_cisuz.jar; db2jcc javax.jar
- Oracle: ojdbc14.jar
- SQL Server JDBC driver provided by Microsoft: sqljdbc.jar
- SQL Server JDBC driver provided by DataDirect: sqlserver.jar; base.jar; util.jar

B. Run the following task. Include each node name as a comma separated list in the command:

Running the task: You do not have to run this task more than once. You can run this task from any node in the cluster.

1. Set the property value for federated.dbDbType in the wkplc.properties file if using a database user registry.
2. Run the ./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=password -DDDbDomain=federated.db -DVmmNodeName=node_name -Db_type.NodeDbLibrary=local full path of the database jars task from the wp_profile_root/ConfigEngine directory on each node to create the variable used to access the VMM database jars. Note: VmmNodeName is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The db_type in db_type .NodeDbLibrary should be set to the type of database you are using, for example db2.

C. Stop and restart all necessary servers to propagate your changes.

7. Run the ./ConfigEngine.sh wp-create-db -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add a database user registry to the default federated repository. Note: Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

9. Perform the following steps to update the user registry where new users and groups are stored:

Note: If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

Attention: During installation, the default file repository creates a default value in the personAccountRdnProperties and groupRdnProperties parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

B. Enter a value for the following required parameters in the wkplc.properties file under the VMM supported entity types configuration heading:

Note: See the properties file for specific information about the required parameters and for advanced parameters.
C. Save your changes to the wkplc.properties file.

D. Run the `./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password task`, from the `wp_profile_root` directory, to delete the old attributes before adding the new attributes.

E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Run the `./ConfigEngine.sh wp-query-repository -DWasPassword=password task`, from the `wp_profile_root` directory, to list the names and types of configured repositories.

**Parent topic:** Configuring the default federated repository on HP-UX
Adding realm support on HP-UX

A realm is a group of users from one or more user registries that form a coherent group within IBM® WebSphere® Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

Before configuring realm support, you must add all LDAP user registries and/or database user registries, that you will use to create a single realm or multiple realms, to the federated repository. If you are going to create multiple realms, you must create all required base entries within your LDAP user registries and/or database user registries. All base entry names must be unique within the federated repository. In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add realm support to your user registry model:

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
2. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - realmName
   - securityUse
   - delimiter
   - addBaseEntry
3. Save your changes to the wkplc.properties file.
4. Run the ./ConfigEngine.sh wp-create-realm -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add a new realm to the Virtual Member Manager configuration. Important: To create multiple realms, ensure that your federated repository contains the required unique base entries. Stop and restart the appropriate servers for your installation environment, and then update the wkplc.properties file with the base entry information and rerun the wp-create-realm task. Repeat these steps until all realms are created.
5. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.
6. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading and then save your changes: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - realmName
   - realm.personAccountParent
   - realm.groupParent
   - realm.orgContainerParent
7. Run the ./ConfigEngine.sh wp-modify-realm-defaultparents -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to update the default parents per entity type and realm. Important: Stop and restart the appropriate servers for your installation environment before rerunning this task for any additional entity types and realms.
8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

9. Optional: Perform the following steps to add additional base entries to the realm configuration; for example, if you had two additional base entries (base entry 1 and base entry 2) to add to the realm you just created, you would update the wkplc.properties file with the information from base entry 1 and then run this task. Then you would update the properties file with the information for base entry 2 and then run this task:
   A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
   B. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading:
      - realmName
      - addBaseEntry
   C. Save your changes to the wkplc.properties file.
   D. Run the ./ConfigEngine.sh wp-add-realm-baseentry - DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add additional LDAP base entries to the realm configuration.
   E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID; this step is required if you change the default realm:
   A. Create a new user in the Manage Users and Groups portlet to replace the current WebSphere Application Server administrative user.
   B. Create a new user in the Manage Users and Groups portlet to replace the current WebSphere Portal administrative user.
   C. Create a new group in the Manage Users and Groups portlet to replace the current group.
   D. Run the ./ConfigEngine.sh wp-change-was-admin-user - DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid task, from the wp_profile_root/ConfigEngine directory, to replace the old WebSphere Application Server administrative user ID and group ID with the new user and group. Important: You must provide the full distinguished name (DN) for the newAdminId and newAdminGroupId parameters.
      Additional parameter for stopped servers: This task verifies the user against a running server instance. If the server is stopped, add the -Dskip.ldap.validation=true parameter to the task to skip the validation.
   E. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.
   F. Run the ./ConfigEngine.sh wp-change-portal-admin-user - DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid task to replace the old WebSphere Portal administrative user ID and group ID with the new user and group. Important: You must provide the full distinguished name (DN) for the newAdminId and newAdminGroupId parameters.
      Additional parameter for stopped servers: This task verifies the user against a running server instance. If the server is stopped, add the -Dskip.ldap.validation=true parameter to the task to skip the validation.
   G. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

11. Optional: Perform the following steps to set the realm you created as the default realm: Remember: Only users defined in base entries that exist in the default realm are able to log into WebSphere Portal. If you find that a user cannot log in to WebSphere Portal, check to see if the base entry that contains the user exists in the default realm. You can run the wp-query-realm-baseentry task to see what base entries are part of the default realm. If the default realm is missing the base entry, run the wp-add-realm-baseentry task to add the base entry to the default realm.
A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. For `defaultRealmName`, type the `realmName` property value you want to use as the default realm.

C. Save your changes to the `wkplc.properties` file.

D. Run the `./ConfigEngine.sh wp-default-realm -DDWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to set this realm as the default realm.

E. Stop and restart all necessary servers to propagate your changes.

12. Optional: Perform the following steps to query a realm for a list of its base entries:

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. For `realmName`, type the name of the realm you want to query.

C. Save your changes to the `wkplc.properties` file.

D. Run the `./ConfigEngine.sh wp-query-realm-baseentry -DDWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to list the base entries for a specific realm.

13. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. Enter a value for `realmName` or leave blank to update the default realm.

C. Save your changes to the `wkplc.properties` file.

D. Run the `./ConfigEngine.sh wp-modify-realm-enable-dn-login -DDWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `./ConfigEngine.sh wp-modify-realm-disable-dn-login -DDWasPassword=password` task to disable the feature.

E. Stop and restart all necessary servers to propagate your changes.

**Parent topic:** Configuring the default federated repository on HP-UX
Adapting the attribute configuration on HP-UX

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

- **Prerequisites**
  - Preparing user registries on HP-UX
  - Choosing your user registry model on HP-UX

After installation, IBM WebSphere Portal has a predefined set of attributes for users and groups. Your LDAP server may have a different set of predefined user and group attributes. To ensure proper communication between WebSphere Portal and your LDAP server, you can configure additional attributes and flag existing attributes as required or unsupported on a per repository basis or for all configure repositories.

LDAP servers can only handle attributes that are explicitly defined in their schema. The LDAP server's schema is different from the WebSphere Portal schema but the two schemas should match for proper communication between WebSphere Portal and the LDAP server. The task to add the LDAP user registry does some basic attribute configurations depending on the type of LDAP server that you choose. You may, however, still need to adapt the WebSphere Portal configuration to match the LDAP schema; for example, if an attribute is defined in WebSphere Portal but not in the LDAP server, you will need to perform one of the following tasks to resolve this mismatch:

- Flag the attribute as unsupported for the LDAP server
- Introduce an attribute mapping that maps the WebSphere Portal attribute to an attribute defined in the LDAP schema

Perform the following tasks to adapt the attribute configuration to match the configured LDAP server(s) and your business needs:

1. **Querying the defined attributes on HP-UX**
   After installing IBM WebSphere Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

2. **Adding attributes on HP-UX**
   The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM WebSphere Portal and your user registry.

3. **Mapping attributes**
   After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

4. **Removing attributes**
   Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

**Parent topic:** Configuring WebSphere Portal to use a user registry on HP-UX

**Previous topic:** Choosing your user registry model on HP-UX

**Next topic:** Stand-alone server: Configuring WebSphere Portal to use dynamic groups on HP-UX
Related tasks

- Adding an LDAP user registry on HP-UX
- Adding an LDAP user registry over SSL on HP-UX
- Configuring a stand-alone LDAP user registry on HP-UX
- Configuring a stand-alone LDAP user registry over SSL on HP-UX
Querying the defined attributes on HP-UX

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

Warning: If you are using a database user registry or a property extension database, copy the database drivers to the AppServer_root/lib directory before executing this script.

Run the `./ConfigEngine.sh wp-query-attribute-config -DWasPassword=yourpassword` task, from the wp_profile_root/ConfigEngine directory, any time during the configuration process or at runtime to query an overview of the currently defined attributes. This task creates the availableAttributes.html report, located in the wp_profile_root/ConfigEngine/log directory. The report contains one table that lists the available attributes for Users (PersonAccount) and one table that lists the available attributes for Groups. For each configured repository there is a column that indicates if the attribute is flagged as unsupported or if the attribute is mapped to a different LDAP attribute. Note: This task does not validate the existence of attributes in the LDAP schema.

Parent topic: Adapting the attribute configuration

Next topic: Adding attributes on HP-UX
Adding attributes on HP-UX

The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM® WebSphere® Portal and your user registry.

- **Prerequisites**
  - Querying the defined attributes on HP-UX

Perform the following steps to add new attributes to your user registry:

1. Install the required Enterprise Archive (.ear) file on WebSphere Application Server. Select the appropriate task to install the .ear file:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>Perform the following steps to install the .ear file in a stand-alone environment: Open a command prompt. Navigate to the <code>wp_profile_root/ConfigEngine</code> directory. Run the <code>./ConfigEngine.sh wp-la-install-ear -DWasPassword=password</code> task.</td>
</tr>
<tr>
<td>Clustered</td>
<td>Perform the following steps to install the .ear file in a clustered environment: Open a command prompt. Navigate to the <code>wp_profile_root/ConfigEngine</code> directory on the primary node. Run the <code>./ConfigEngine.sh wp-la-install-ear -DWasPassword=dmgr_password -DServerName=dmgr -DNServerName=node_name</code> task. Where <code>node_name</code> is the name of the node where the deployment manager resides; you can find the <code>node_name</code> value in the WebSphere Application Server Administrative Console under System administrator &gt; Deployment Manager &gt; Runtime tab &gt; General Properties &gt; Node Name.</td>
</tr>
</tbody>
</table>

2. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

3. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

4. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Property Extension Properties heading: **Note**: See the properties file for specific information about the required parameters and for advanced parameters.
   - `la.providerURL`
   - `la.propertyName`
   - `la.entityTypes`
   - `la.dataType`
   - `la.multiValued`
5. Save your changes to the `wkplc.properties` file.

6. Run the `./ConfigEngine.sh wp-add-property -DWasPassword=password` task to add the attribute to the user registry.

   **Note:** This task performs an EJB call to WebSphere Application Server, which must authenticate against WebSphere Application Server. Depending on the configuration in the `sas.client.props` file, you may receive a popup window or a command line prompt asking for user identity and password. Enter the WebSphere Application Server user ID and password.

   **Remember:** If you have multiple properties to add, repeat all steps, except for the `wp-la-install-ear` task, until all new attributes are added.

7. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents.*

**Related tasks**
- Adapting the attribute configuration
- Querying the defined attributes on HP-UX
- Mapping attributes

**Related topics**
- Starting and stopping servers, deployment managers, and node agents
Mapping attributes

After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

- **Prerequisites**
  - Querying the defined attributes on HP-UX
  - Adding attributes on HP-UX

Perform the following steps to map attributes between WebSphere Portal and your LDAP server; if you have multiple LDAP servers, you will need to perform these steps for each LDAP server:

1. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
2. Enter a value for one of the following sets of parameters in the `wkplc.properties` file to identify your LDAP server:
   - **Note**: Make sure you use the same values you used to configure your LDAP server.

<table>
<thead>
<tr>
<th>Table 1. Identifying your LDAP server in the wkplc.properties file.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Repository type</strong></td>
</tr>
<tr>
<td>Stand-alone</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Federated</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

3. Run one of the following tasks to check that all defined attributes are available in the configured LDAP user registry:
   - **Task**: See the properties file for specific information about the required parameters and for advanced parameters.

<table>
<thead>
<tr>
<th>Table 2. Task to check that all defined attributes are available in the configured LDAP user registry.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Repository type</strong></td>
</tr>
<tr>
<td>Stand-alone</td>
</tr>
<tr>
<td>Federated</td>
</tr>
</tbody>
</table>
4. Open the ConfigTrace.log file, located in the \wp_profile_root\ConfigEngine\log directory, to review the following output for the PersonAccount and Group entity type:

- **The following attributes are defined in WebSphere Portal but not in the LDAP server**
  - This list contains all attributes that are defined in WebSphere Portal but not available in the LDAP. Flag attributes that you do not plan to use in WebSphere Portal as unsupported. Map the attributes that you plan to use to the attributes that exist in the LDAP; you must also map the \_uid, cn, firstName, sn, preferredLanguage, and ibm-primaryEmail\ attributes if they are contained in the list.

- **The following attributes are flagged as required in the LDAP server but not in WebSphere Portal**
  - This list contains all attributes that are defined as "MUST" in the LDAP server but not as required in WebSphere Portal. You should flag these attributes as required within WebSphere Portal; see the step below about flagging an attribute as either unsupported or required.

- **The following attributes have a different type in WebSphere Portal and in the LDAP server**
  - This list contains all attributes that WebSphere Portal might ignore because the data type within WebSphere Portal and within the LDAP server do not match.

5. Use a text editor to open the wkplc.properties file, located in the \wp_profile_root\ConfigEngine\properties directory.

6. Enter a value for one of the following sets of parameters in the wkplc.properties file to correct any issues found in the config trace file: Table 3. Parameters to define in the wkplc.properties file to correct any issues found in the config trace file.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stand-alone</strong></td>
<td>The following parameters are found under the LDAP attribute configuration heading: <strong>Note:</strong> See the properties file for specific information about the required parameters and for advanced parameters. standalone.ldap.idstandalone.ldap.attributes.nonSupported.standalone.ldap.attributes.nonSupported.deletestandalone.ldap.attributes.mapping.ldapName=estandalone.ldap.attributes.mapping.portalName=estandalone.ldap.attributes.mapping.entityTypes=For example, the following values will flag certificate and members as unsupported attributes and will map ibm-primaryEmail to mail and ibm-jobTitle to title for both the PersonAccount and Group entityTypes: standalone.ldap.attributes.nonSupported=certificate, members standalone.ldap.attributes.nonSupported.delete=standalone.ldap.attributes.mapping.ldapName=mail, title standalone.ldap.attributes.mapping.portalName=ibm-primaryEmail, ibm-jobTitle standalone.ldap.attributes.mapping.entityTypes=PersonAccount, Group</td>
</tr>
</tbody>
</table>
The following parameters are found under the VMM Federated repository properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

- `federated.ldap.attributes.nonsupported`
- `federated.ldap.attributes.nonsupported.delete`
- `federated.ldap.attributes.mapping.ldapName`
- `federated.ldap.attributes.mapping.portalName`
- `federated.ldap.attributes.mapping.entityTypes`

For example, the following values will flag certificate and members as unsupported attributes and will map `ibm-primaryEmail` to `mail` and `ibm-jobTitle` to `title` for both the PersonAccount and Group entityTypes:

<table>
<thead>
<tr>
<th>federated.ldap.attributes.nonsupported=</th>
<th>certificate, members</th>
</tr>
</thead>
<tbody>
<tr>
<td>federated.ldap.attributes.nonsupported.delete=</td>
<td>certificate, members</td>
</tr>
<tr>
<td>federated.ldap.attributes.mapping.ldapName=</td>
<td>mail, title</td>
</tr>
<tr>
<td>federated.ldap.attributes.mapping.portalName=</td>
<td>ibm-primaryEmail, ibm-jobTitle</td>
</tr>
<tr>
<td>federated.ldap.attributes.mapping.entityTypes=</td>
<td>PersonAccount, Group</td>
</tr>
</tbody>
</table>

7. Save your changes to the `wkplc.properties` file.

8. Run one of the following tasks to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between WebSphere Portal and the LDAP user registry:

   **Table 4. Task to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between Portal and the LDAP user registry.**

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td><code>./ConfigEngine.sh wp-update-standalone-ldap-attribute-config -DWasPassword=password</code> task, from the <code>wp_profile_root/ConfigEngine</code> directory</td>
</tr>
<tr>
<td>Federated</td>
<td><code>./ConfigEngine.sh wp-update-federated-ldap-attribute-config -DWasPassword=password</code> task, from the <code>wp_profile_root/ConfigEngine</code> directory</td>
</tr>
</tbody>
</table>

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents.*

10. Optional: Perform the following steps to flag an attribute as either unsupported or required for the entire WebSphere Portal environment instead of just for the specified LDAP:

   A. Enter a value for the following required parameters in the `wkplc.properties` file: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - `user.attributes.required`
      - `user.attributes.nonsupported`

   B. Save your changes to the `wkplc.properties` file.

   C. Run the `./ConfigEngine.sh wp-update-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory.

   D. Stop and restart all necessary servers to propagate your changes.

**Parent topic:** Adapting the attribute configuration

**Previous topic:** Adding attributes on HP-UX

546
Next topic: Removing attributes

Related tasks
Starting and stopping servers, deployment managers, and node agents
Removing attributes

Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

- Prerequisites
  - Querying the defined attributes on HP-UX
  - Adding attributes on HP-UX
  - Mapping attributes

Perform the following steps to remove an attribute from your database: Important: Do not remove attributes that have already been populated with user values because this can cause database inconsistencies.

Cluster note: In a clustered environment, perform the following steps on the deployment manager and then resynch the nodes.

1. Perform the following steps to remove an attribute stored in a property extension database; this is an attribute that was added using the wp-add-la-property task.
   A. Open the tool you use to edit your database.
   B. Verify that your attribute name is available in the LAPROP table.
   C. Delete the required attributes from the LAPROP table.
   D. Open the wimxmlextension.xml file, located in the wp_profile_root/config/cells/cellname/wim/model directory.
   E. Locate and delete the propertySchema definition for the attributes that you deleted from the LAPROP table; for example:
      ```
      <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String"
       multiValued="true" propertyName="attribute_name">
        <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
      </wim:propertySchema>
      ```
   F. Save your changes to the wimxmlextension.xml file.
   G. Open the wimconfig.xml file, located in the wp_profile_root/config/cells/cellname/wim/config directory.
   H. Locate and delete the propertiesNotSupported definitions for the attributes that you deleted from the LAPROP table; for example:
      ```
      <config:propertiesNotSupported name="attribute_name">
      ```
   I. Save your changes to the wimconfig.xml file.
   J. Stop and restart the server1 and WebSphere_Portal servers from the wp_profile_root/bin directory.

2. Perform the following steps to remove an attribute that is not stored in a property extension database:
   A. Open the wimxmlextension.xml file.
   B. Locate and delete the propertySchema definition for the attributes you previously added:
      ```
      <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String"
       multiValued="true" propertyName="attribute_name">
        <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
      </wim:propertySchema>
      ```
   C. Save your changes to the wimxmlextension.xml file.
   D. Open the wimconfig.xml file.
   E. Locate and delete the stanza that corresponds to the custom attribute you deleted from the wimextension.xml file; for example:
      ```
      <config:attributes name="attribute_name" propertyName="property_name">
      ```
   F. Save your changes to the wimextension.xml file.

548
F. Save your changes to the `wimconfig.xml` file.
G. Stop and restart the server1 and WebSphere_Portal servers.

**Parent topic:** Adapting the attribute configuration  
**Previous topic:** Mapping attributes
Stand-alone server: Configuring WebSphere Portal to use dynamic groups on HP-UX

By default, WebSphere Portal is enabled for static groups. However, the Virtual Member Manager (VMM) allows users to be members of either static or dynamic groups. Static groups are those where a persistent binding exists between a group and its members. Dynamic groups are those where a search query is defined to retrieve the members of a group. If you have your LDAP server configured to use dynamic groups, complete the steps in this task for WebSphere Portal to use dynamic group queries when you setup your LDAP server.

Perform the required tasks to configure either a stand-alone or federated LDAP server security.

- **Prerequisites**
  - Preparing user registries on HP-UX
  - Choosing your user registry model on HP-UX
  - Adapting the attribute configuration

The steps in this task use `groupOfURLs` as the object class for dynamic groups and `memberURL` as the dynamic membership attribute. The actual values for object classes and dynamic membership attributes can vary depending on your LDAP server. For this reason, you should export an LDIF file to verify the object classes and dynamic membership attributes. Either refer to your LDAP documentation or ask your LDAP administrator for instructions on exporting an LDIF file.

**Clustered environments:** Perform the following steps on the Deployment Manager then synchronize the nodes.

To configure WebSphere Portal to use dynamic groups, do the following:

1. Choose the appropriate set of steps, depending on your LDAP server environment: *Table 1. Steps for enabling dynamic groups*

<table>
<thead>
<tr>
<th>LDAP server environment</th>
<th>Steps to perform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone LDAP server or federated LDAP server(s)</td>
<td>Navigate to the following directory: <code>wp_profile_root/cells/cell_name/wim/config</code>&lt;br&gt;Locate and open <code>wimconfig.xml</code> with any text editor.&lt;br&gt;Add the following line to the <code>&lt;config:groupConfiguration&gt;</code> tag:&lt;br&gt;<code>&lt;config:dynamicMemberAttributes name=&quot;memberurl&quot; objectClass=&quot;groupofurls&quot;/&gt;</code>&lt;br&gt;Save and close <code>wimconfig.xml</code>.</td>
</tr>
</tbody>
</table>
2. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

### Federated LDAP server(s)

| Log in to the administration console. Select **Security > Secure administration, applications, and infrastructure**. Under **Available realm definitions**, select **Federated repositories** and click **Configure**. Under **Related Items**, click **Manage repositories**. Select the appropriate repository from the list. Under **Additional Properties**, click **Group attribute definition** then click **Dynamic member attributes**. Click **New** and specify values for the **Name** and **Object class** fields as appropriate. For example, **Name**: `memberurl`, **Object class**: `groupofurls`. Click **OK** and save the changes to the master configuration. |

**Parent topic:** Configuring WebSphere Portal to use a user registry on HP-UX

**Previous topic:** Adapting the attribute configuration

**Next topic:** Stand-alone server: Enabling referrals for your LDAP user registry on HP-UX
Stand-alone server: Enabling referrals for your LDAP user registry on HP-UX

Referrals redirect object requests from one LDAP server to another when objects do not exist or cannot be located in a particular directory tree. You should enable referrals if your environment has more than one user registry existing on multiple servers or domains.

- **Prerequisites**
  - Preparing user registries on HP-UX
  - Choosing your user registry model on HP-UX
  - Adapting the attribute configuration
  - Stand-alone server: Configuring WebSphere Portal to use dynamic groups on HP-UX

To configure your portal to use LDAP referrals, do the following:

1. Use any text editor to open the `wkplc.properties` file in the following directory: `wp_profile_root/ConfigEngine/properties`.
   
   2. Specify values for the following parameters:
      - `et.ldap.id` = ID_of_your_LDAP_server
      - `et.ldap.host` = hostname_of_your_LDAP_server
      - `et.ldap.referral` = follow
   
   3. Save and close `wkplc.properties`.

   4. Run the following task from the `wp_profile_root/ConfigEngine` directory to create an LDAP entity type:
      - Windows: `ConfigEngine.bat wp-update-et-ldap -DWasPassword=password`
      - IBM® i5/OS: `ConfigEngine.sh wp-update-et-ldap -DWasPassword=password`

5. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents.*

**Parent topic:** Configuring WebSphere Portal to use a user registry on HP-UX  
**Previous topic:** Stand-alone server: Configuring WebSphere Portal to use dynamic groups on HP-UX
Tune your servers

Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

- Prerequisites
  - Preparing your HP-UX operating system
  - Installing WebSphere Portal on HP-UX
  - Configuring databases
  - Configuring WebSphere Portal to use a user registry on HP-UX
  - Base Portal Tuning Scenarios

Tuning a WebSphere Portal environment involves tuning and configuring the various systems and components of the environment. The tuning guide provides general concepts and details specifics configuration instructions. Instructions are included for:
- Configuring the application server and the resources defined for that application server
- Determining the cloning strategy for expanding or extending the environment
- Tuning the database(s) and database server
- Tuning the directory server and its database
- Tuning the Web server
- Tuning the operating system and network
- Tuning the WebSphere Portal services

Parent topic: Setting up a stand-alone server on HP-UX
Previous topic: Configuring WebSphere Portal to use a user registry on HP-UX

Related information
- WebSphere Portal and Lotus Web Content Management 6.1.x Performance Tuning Guide
- IBM WebSphere Portal Performance Troubleshooting Guide
Setting up a stand-alone server on i5/OS

After you have prepared the operating system you are ready to install WebSphere Portal.

1. **Prerequisites**

   - Technotes for installation and configuration issues

   Select the operating system on which you are installing WebSphere Portal.

2. **Preparing your IBM i5/OS**

   View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

3. **Installing WebSphere Portal on IBM i5/OS**

   IBM WebSphere Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

4. **Preparing DB2 for i5/OS for a stand-alone production server**

   To setup a remote DB2 for i5/OS database, create user IDs and databases on a remote server. Tasks are provided to assist with creating the user IDs and the databases. Before you can use the tasks, you must modify properties files.

5. **Preparing the Web server when portal is installed on IBM i5/OS**

   Install and configure the Web server plug-in, provided by IBM WebSphere Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

6. **Configuring WebSphere Portal to use a user registry on IBM i5/OS**

   Configure user registry security on IBM WebSphere Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

6. **Tune your servers**

   Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

**Parent topic:** Setting up a stand-alone production server
Preparing your IBM i5/OS

View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

WebSphere Portal can be installed locally or remotely using a Windows workstation.

You need the following information to install remotely:
- A supported version of Microsoft® Windows®
- Workstation CD-ROM drive
- TCP/IP connection to the IBM i5/OS system where you will install WebSphere Portal
- The IBM i5/OS server must be in an unrestricted state
- A valid user ID and password on the IBM i5/OS system
- A user profile with a user type (user class) of *SECOFR (other than QSECOFR) to install and configure WebSphere Portal

You need the following information to install locally:
- IBM i5/OS CD-ROM drive
- The IBM i5/OS server must be in an unrestricted state
- A valid IBM i5/OS user ID and password
- A user profile with a user type (user class) of *SECOFR (other than QSECOFR) to install and configure WebSphere Portal
Installing WebSphere Portal on IBM i5/OS

IBM® WebSphere® Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

Review the Supported hardware and software requirements for your operating system and ensure that all hardware and software matches the minimum product level before installing WebSphere Portal. Review Installation methods, options, and sources before installing to determine your installation source location and installation method. **Note:** If you plan to use IBM WebSphere Application Server Version 7.0, you must purchase and install it before installing WebSphere Portal on the existing WebSphere Application Server Version 7.0 instance.

- **Prerequisites**
  - Preparing your IBM i5/OS

Perform the following steps to install WebSphere Portal:

1. Type `ping yourserver.yourcompany.com` on a command line to verify that your fully qualified host name is properly configured.
2. Type `ping localhost` on a command line to verify that your network is properly configured.
3. Setup a static IP address on the server where you will install WebSphere Portal.
4. If you are installing with an existing WebSphere Application Server instance, ensure it is installed at the supported level and has the following features installed:
   - **Restriction:** Installing WebSphere Portal Version 6.1.0.1 or higher on an existing WebSphere Application Server Version 7.0 instance where WebSphere Portal Version 7.0 was installed is unsupported.
   - **Important:** Besides ensuring that the existing WebSphere Application Server instance is at the supported level, you will also need to ensure that Apache Derby is installed at the supported level before installing WebSphere Portal. See WebSphere Portal detailed system requirements for supported levels.

- Application Server
- Administration
  - Scripted Administration
  - Administrative Console
- Ant and Deployment Tools
  - Deploy Tool
  - Ant Utilities

5. Choose one of the following installation commands based on the installation method you want to use to install WebSphere Portal; see Installation Methods for more information:
   - **Advance installation parameters:** To customize your installation, you can add various parameters to your installation commands; for example, you can specify your port information. See Advanced installation parameters for information.
   - **Restriction:** When defining your installation path, the ONLY supported characters are ASCII letters [A-Z and a-z], numbers [0-9], slashes [/ or \, depending on your operating system], and the dash [-].

---

**Table 1. Installation task options**

<table>
<thead>
<tr>
<th>Installation type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

556
6. Verify your installation was successful; access WebSphere Portal using the `http://yourserver:yourport/wps/portal` format. Run the `ConfigEngine.sh list-server-ports -DWasPassword=password` task from the `wp_profile_root/ConfigEngine` directory to generate the `wp_profile_root/ConfigEngine/log/wp_PortMatrix.txt` file that lists the WebSphere Portal ports for your installation.

7. Optional: After the WebSphere Portal installation completes successfully, run the `ConfigEngine.sh configure-express -DPortalAdminPwd=password -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, if you want to enable the sample IBM Lotus Web Content Management content, such as internet and intranet sample sites.

   **Restriction:** Run the `configure-express` task before configuring your database, user registry, context root, security, etc. If you ran any tasks other than the install task, do not run this task.

   **Note:** See [http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_ovw.html](http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_ovw.html) for tutorials on how to use the sample content.

   The sample content includes:

   - Creates a group called `contentAuthors`; members of this group are given privileges to create content in the sample Internet and intranet sites. Navigate to the **Administration** area and then click **Access > Users and Groups**.

   - Creates two new Web Content Management Libraries: "Internet Web Content 6.1.0" and "Intranet Web Content 6.1.0". Navigate to the **Administration** area and then click **Portal Content > Web Content Libraries**.
- Adds a portlet filter and applies the filter to various portlets in the sample Internet and intranet sites. You can see the definition of the filter in the WebSphere Application Server Administration console and examining the custom resources under the **Environment** area.

- Creates two new theme policies: InternetStyle and IntranetStyle. These styles are applied to sample Internet and intranet sites. Navigate to **Theme Customizer** and then select the style.

- Creates several portlet clones of the Web Content Management rendering portlet. These portlet clones are used on sample Internet and intranet sites.

- Creates two virtual portals with context roots of wps/portal/intranet and wps/portal/internet. These are the sample Internet and intranet sites. Go to http://yourserver:yourport/wps/portal/internet and http://yourserver:yourport/wps/portal/intranet to access them.

- Creates several sample credential slots, including "Default slot for E-mail", "Default slot for Feeds", "Default slot for Miscellaneous", "Default slot for Web Clipping", and "Default slot for Web Content Management". Navigate to the **Administration** area and then click **Access > Credential Vault > Manage System Vault Slots**.

- Modifies the portlet palette to include some new portlets. Click the **Portlets** link to see the portlet palette.

8. Optional: If you ran the **configure-express** task, the owner of the items in the Web content libraries containing the Internet and Intranet Site Template content will be listed as uid=xyzadmin,o=defaultWIMFileBasedRealm. To update the owner information for these items to correspond to your portal administrator ID, complete the following steps:

A. Edit the `wp_profile_root/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties` file.

B. Add the following line to the file:

```
uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
```

Replace **portal_admin_DN** with the distinguished name of the portal administrator.

C. Save your changes and close the file.

D. Run the `ConfigEngine.sh action-express-memberfixer -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory.

If you installed WebSphere Portal on an i5/OS Version 6.1 system, install the **MF45016** fix; see Fix Central to access the fix. Set the following two cache properties under the WP_CacheManagerService resource environment provider in the WebSphere Application Server Administrative Console; see Setting service configuration properties for information:

- `cacheglobal.softref=true`
- `cacheglobal.lifetime=1200`

Before you start the WebSphere Application Server server, you need to configure the software license agreement to set the usage limit from the Proof of Entitlement (POE) or invoice; see Configuring software license information for details.
Advanced installation parameters
Preparing DB2 for i5/OS for a stand-alone production server

To setup a remote DB2 for i5/OS database, create user IDs and databases on a remote server. Tasks are provided to assist with creating the user IDs and the databases. Before you can use the tasks, you must modify properties files.

**Prerequisites**
- Preparing your IBM i5/OS
- Installing WebSphere Portal on IBM i5/OS

1. **Preparing for creation of databases**
   View the mandatory preparation tasks prior to creating databases that are required by WebSphere Portal.

2. **Create user profiles**
   View information on setting up user profiles for DB2 for i5/OS to work with WebSphere Portal.

3. **Creating remote databases**
   A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

4. **Modifying database properties**
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. **Transferring DB2 for i5/OS manually**
   View the steps to manually transfer data to the IBM DB2 Universal Database™ for iSeries® database you have set up.
   As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

6. **Verifying database connections**
   After WebSphere Portal is configured to work with your database, test the database connection to ensure that it operates correctly.

**Parent topic:** Setting up a stand-alone server on i5/OS

**Previous topic:** Installing WebSphere Portal on IBM i5/OS

**Next topic:** Preparing the Web server when portal is installed on IBM i5/OS
Preparing for creation of databases

View the mandatory preparation tasks prior to creating databases that are required by WebSphere Portal.

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

Tip: For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Preparing database domain property values prior to creating database names and schemas. Before you manually configure your database schemas, you must first edit your database domain properties files.

A. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

B. The WebSphere Portal database can be used to hold information for applications such as Personalization (Feedback) and LikeMinds. To prepare the database to hold such application information, you should use similar naming conventions for property values such as release.DbName. Here are some examples:
   - release.DbName=hostname/WP61REL
   - community.DbName=hostname/WP61COM
   - customization.DbName=hostname/WP61CUS
   - jcr.DbName=hostname/WP61JCR
   - feedback.DbName=hostname/WP61FBK
   - likeminds.DbName=hostname/WP61LKM

   When you create a schema, you must use the following schema naming conventions on the i5/OS system: Note: The default schema names may be used with the product.
   - Length cannot exceed 10 characters
   - All alphanumerical characters are allowed ("A" through "Z" and "1" through "0")
   - The following characters are invalid:
     - spaces
     - null values
     - asterisk (*)
     - quotation marks ("")
     - colon ():
     - greater than symbol (>)
     - less than symbol (<)
     - vertical bar (|)
Notes:
- Make sure you know what valid schema names are and do not use a schema name which already exists on the local or remote system. Follow the documentation of the target database management system in order to define a valid schema name as restrictions apply. Note that the Create WebSphere Portal wizard will automatically check schema names for you.
- For more information on database and schema naming conventions, refer to DB2 Universal Database for System i5 topic in the System i5 information center.

- Do not change any settings other than those that are specified in these substeps.
- If you are using a remote database, be sure to enter the values for the remote server.
- Use / instead of \ for i5/OS.
- Some values, shown here in italics, might need to be modified to your specific environment.
- Password considerations: For security reasons, you should not store passwords in the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files. It is recommended that you edit each of the properties files prior to running a configuration task, inserting the passwords needed for that task. Then, after the task has run, you should delete all passwords from each file.
- There might be additional database properties other than those listed here. Only change the properties within this table; skip all other properties.
- Depending on which database domain has to be configured, the variable dbdomain may need to be replaced by:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds

C. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment. You can also modify each properties file locally on your System i5 system by typing the following on an OS/400 command line in a 5250 session: EDTF 'wp_profile_root/ConfigEngine/properties/property filename.properties'

where property filename is wkplc_comp, wkplc, or wkplc_dbtype.

Note: You must have a user profile on the System i5 server and must have at least *USE special authority to edit the properties file.

Tip: The steps for transferring data to another supported database section provide instructions for manually transferring data. Instead of performing the following steps, you can use the configuration wizard, which is a graphical user interface, to transfer data to another supported database.

The following properties must be changed before creating a database name and schema on a local or remote System i5 server.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.

A. For dbdomain.DbType, type db2_iseries.

B. For dbdomain.DbName, type the name of the WebSphere Portal domain database. Note: This value is also the database element in the dbdomain.DbUrl property.

C. For dbdomain.DbSchema, type the schema name of the database domain. Note: Review your target database management system documentation to define a valid schema name. Some database management systems have
schema name restrictions that you need to understand.

D. For `dbdomain.DataSourceName`, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
   - releaseDS
   - communityDS
   - customizationDS
   - jcrDS
   - lmdbDS
   - feedback

E. For `dbdomain.DbUrl`, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:** The database element of this value should match the value of DbName.

F. For `dbdomain.DbUser`, type the user ID for the database administrator.

G. For `dbdomain.DbPassword`, type the password for the database administrator.

H. For `dbdomain.DBA.DbUser`, type the database administrator user ID for privileged access operations during creation of the database.

I. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.

3. Save and close the file.

4. Update the following properties in the file `wkplc_dbtype.properties`.
   A. For `db2_iseries.DbDriver`, type the name of the class that `SqlProcessor` uses to import SQL files.
   B. For `db2_iseries.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.
   C. For `db2_iseries.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.
   D. For `db2_iseries.DbDriverType`, type the number representing the driver type for the database.

5. Save and close the file.

6. Update the following property in the file `wkplc.properties`.
   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

7. Save and close the file.

**Parent topic:** Preparing DB2 for i5/OS for a stand-alone production server  
**Next topic:** Create user profiles
Create user profiles

View information on setting up user profiles for DB2 for i5/OS to work with WebSphere Portal.

Before you begin:

- The user profile for the database owner should be different from the administrator user profile used to perform the installation. The administrator user profile may have more authority than is required and usually belongs to an individual, whereas the database user profile may have minimal authority and can be shared.

- Create a database user profile that does not require a password change over a period of time. If the password for the database user profile changes, WebSphere Portal must be re-configured to use the new password.

- Create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

- **Prerequisites**
  - Preparing for creation of databases

To create user profiles, follow the instructions that are provided with the DB2 for i5/OS documentation.

**Parent topic:** Preparing DB2 for i5/OS for a stand-alone production server  
**Previous topic:** Preparing for creation of databases  
**Next topic:** Creating remote databases
Creating remote databases

A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

Before you begin:
- The user ID and password used must have the authority to create database libraries on the remote System i5 machine.
- For every property instance of a database that uses *LOCAL/schema, replace it with HostName/schema. For example, the default database and database library name for WebSphere Portal release domain is release.DbName=wpsdb. If you wanted to create this database library on a remote database, change the default value to release.DbName=hostname/wpsdb

- Prerequisites
  - Preparing for creation of databases
  - Create user profiles

To create all the domain database libraries, perform the following steps:

1. Start a 5250 session on the remote database machine.
2. Type the i5/OS command WRKRDBDIRE to display the Relational Database Directory Entry for Remote Location *LOCAL and make a note of the value displayed.
3. Sign off from the 5250 session.
4. Start a 5250 session on the local machine where WebSphere Portal is installed.
5. Create a Relational Database Directory Entry on the local system for the remote system using i5/OS command WRKRDBDIRE.
6. Add an entry with the following values:
   - **Relational database**
     - The remote relational database. Use the value noted from the prior step.
   - **Relational database alias**
     - The hostname. Use the short TCP/IP hostname of the remote system
   - **Remote location**
     - The domain qualified hostname. Use the full TCP/IP hostname of the remote system
   - **Type**
     - IP
   - **Port number or service name**
     - DRDA
   - **Remote authentication method**
     - Preferred method: ENCRYPTED
     - Allow lower authentication: ALWLOWER
7. Create the required DB2 packages on the remote database machine by running the following command from the local machine:

```
JAVA CLASS(com.ibm.db2.jdbc.app.DB2PackageCreator) PARM('rdb_alias' 'userid' 'password') PROP((jdbc.drivers 'com.ibm.as400.access.AS400JDBCDriver'))
```

where `rdb_alias` matches the name of the Relational Database Entry you created in step 2, where `userid` is the database administrator user ID on the remote machine, and where `password` is the database administrator password on the remote machine. The output should be:

```
Java program completed
```

Portal, Version 6.1.5
Operating systems: AIX, HP-UX, i5/OS, Linux, Solaris, Windows
8. Press F3 to exit Java Shell Display.
9. Sign off from the 5250 session.
10. Start a 5250 session on the remote database machine.
11. Verify the required DB2 packages were created by running the command `WRKOBJ OBJ(QGPL/QSQCL*) OBJTYPE(*SQLPKG)`

<table>
<thead>
<tr>
<th>Opt</th>
<th>Object</th>
<th>Type</th>
<th>Library</th>
<th>Attribute</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QSQCLIPKGA</td>
<td>*SQLPKG</td>
<td>QGPL</td>
<td>PACKAGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>QSQCLIPKGC</td>
<td>*SQLPKG</td>
<td>QGPL</td>
<td>PACKAGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>QSQCLIPKGL</td>
<td>*SQLPKG</td>
<td>QGPL</td>
<td>PACKAGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>QSQCLIPKGN</td>
<td>*SQLPKG</td>
<td>QGPL</td>
<td>PACKAGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>QSQCLIPKGS</td>
<td>*SQLPKG</td>
<td>QGPL</td>
<td>PACKAGE</td>
<td></td>
</tr>
</tbody>
</table>

12. Start a 5250 session on the local machine where WebSphere Portal is installed.
13. On the command line, enter the following to change directories: `cd wp_profile_root/ConfigEngine`.
14. Press Enter.
15. Change the property values in the configuration properties files before entering the following on the command line:

```
    ConfigEngine.sh create-database
```
16. Press Enter.

**Parent topic:** Preparing DB2 for i5/OS for a stand-alone production server  
**Previous topic:** Create user profiles  
**Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Preparing for creation of databases
  - Create user profiles
  - Creating remote databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**

- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as **release.DbName**, **jcr.DbName**, **feedback.DbName**, and **likeminds.DbName**. For example:
  - `release.DbName=release`
  - `jcr.DbName=jcrdb`
  - `feedback.DbName=fdbkdb`
  - `likeminds.DbName=lmdb`
  - `community.DbName=commdb`
  - `customization.DbName=custdb`

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.

Depending on which database domain has to be configured, replace **dbdomain** with:
- `release`
- `customization`
- `community`
- `jcr`
- `feedback`
- `likeminds`

- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
  - `dbdomain.DbType`
- dbdomain.DbName
- dbdomain.DbUrl
- dbdomain.DbSchema

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.
   A. For dbdomainDbType, type db2_iseries.
   B. For dbdomain_DbName, type the name of the WebSphere Portal domain database. Note: This value is also the database element in the dbdomain_DbUrl property.
   C. For dbdomain_DbSchema, type the schema name of the database domain. Note: Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   D. For dbdomain_DataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For dbdomain_DbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. Note: The database element of this value should match the value of DbName.
   F. For dbdomain_DbUser, type the user ID for the database administrator.
   G. For dbdomain_DbPassword, type the password for the database administrator.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. Important: The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.
   A. For source.dbType, type the database you are currently configured to use. The value for source.dbType is Derby by default.
   B. For source.DbName, type the name of the database domain you are currently using.
   C. For source.DbSchema, type current schema identifier for objects within the database for this domain.
   D. For source.DataSourceName, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.
E. For source.domain.DbUrl, type the url currently used to access your database.
F. For source.domain.DbUser, type the name of the user accessing this database.
G. For source.domain.DbPassword, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file wkplc_dbtype.properties.
   A. For db2_iseries.DbDriver, type the name of the class that SqlProcessor uses to import SQL files.
   B. For db2_iseries.DbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.
   C. For db2_iseries.JdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file wkplc.properties.
   A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

Parent topic: Preparing DB2 for i5/OS for a stand-alone production server
Previous topic: Creating remote databases
Next topic: Transferring DB2 for i5/OS manually
Transferring DB2 for i5/OS manually

View the steps to manually transfer data to the IBM® DB2 Universal Database™ for iSeries® database you have set up. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

**- Prerequisites**
- Preparing for creation of databases
- Create user profiles
- Creating remote databases
- Modifying database properties

**Steps for transferring data to another supported database**

1. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td>stopServer server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td>stopServer WebSphere_Portal -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

2. Validate configuration properties using the following commands:

```bash
ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
```

```bash
ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
```

3. Transfer the database:**Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

**A.** Enter the following command:

```bash
ConfigEngine.sh database-transfer -DWasPassword=password
```

**Note:**
- To select specific database domains to transfer, modify the -DTransferDomainList specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:

```bash
database-transfer -DTransferDomainList=jcr -DWasPassword=password
```

- Use SBMJOB to submit the Qshell script as a batch job to run in *BASE pool when *INTERACT pool does not have 1GB or more of allocated memory. For example:

```bash
SBMJOB CMD(STRQSH CMD(ConfigEngine.sh database-transfer -DWasPassword=password))
```

**B.** After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.
4. Enter the following command to start the WebSphere Portal server:

```
startServer WebSphere_Portal
```

If you have performed database transfer in a clustered environment, you must explicitly copy the `wp_profile_root`/PortalServer/jcr/lib/com.ibm.icm/icm.properties file from the primary node on which the `database-transfer` task was processed to all secondary nodes. This ensures that the secondary nodes have the new JCR database values specified in the `icm.properties` file.

1. Stop the portal server on the secondary node.
2. Copy the `wp_profile_root`/PortalServer/jcr/lib/com.ibm.icm/icm.properties file from the primary node and replace the `icm.properties` file on the secondary node with the new file from the primary node.
3. Start the portal server on the secondary node.

**Parent topic:** Preparing DB2 for i5/OS for a stand-alone production server  
**Previous topics:** Modifying database properties  
**Next topic:** Verifying database connections
Verifying database connections

After WebSphere Portal is configured to work with your database, test the database connection to ensure that it operates correctly.

**Prerequisites**

- Preparing for creation of databases
- Create user profiles
- Creating remote databases
- Modifying database properties
- Transferring DB2 for i5/OS manually

You can verify the connection from a browser or from a command line. To verify that WebSphere Portal is running from a browser, open the portal in a Web browser: http://hostname.yourco.com:port_number/wps/portal, where hostname.yourco.com is the fully qualified host name of the machine where WebSphere Portal is running and port_number is the transport port that is created by IBM® WebSphere® Application Server.

There may be an error if any of the following conditions appear.

- When trying to access the portal you get a 503 error.
- If you had any locale problems with your database, you could see invalid characters, such as ?????, after logging in. This may happen if the character set of the database is not UTF-8 compliant.
- If something went wrong with the data that was transferred, you may not be able to login. WebSphere Portal will indicate you entered an invalid user ID and password even though you know it is valid.

Verify the connection from a command line by completing the following steps:

1. Start a 5250 session on the local machine where WebSphere Portal is installed.
2. For WebSphere Portal on WebSphere Application Server (UserData path), enter the following on the command line: `cd wp_profile_root/ConfigEngine`
3. Enter the following command `ConfigEngine.sh validate-database-connection -DTransferDomainList=release,community,customization,jcr,feedback,likeminds -DWasPassword=password`

For security reasons, you should not leave passwords in the wkplc_comp.properties file. Edit the file prior to running a configuration task and insert the passwords that are needed for that task. After the task has run, delete all passwords from the file.

Alternatively, you can specify the password on the command line rather than update the wkplc_comp.properties file. For example: `ConfigEngine.sh -DPortalAdminPwd=password -DWasPassword=password validate-wps-admin-login`

When installing WebSphere Portal, the passwords in the wkplc_comp.properties file are automatically removed after configuration.

**Parent topic:** Preparing DB2 for i5/OS for a stand-alone production server

**Previous topic:** Transferring DB2 for i5/OS manually
Preparing the Web server when portal is installed on IBM i5/OS

Install and configure the Web server plug-in, provided by IBM® WebSphere® Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

- **Prerequisites**
  - Preparing your IBM i5/OS
  - Installing WebSphere Portal on IBM i5/OS
  - Preparing DB2 for i5/OS for a stand-alone production server

Perform the following steps to install and configure your Web server:

1. Install and configure the Web server; refer to the Web server documentation for information.
2. If using IBM Lotus® Domino®, edit the **NOTES.INI** file on the Web server. Set the **HTTPEnableConnectorHeaders** parameter to 1.
3. If using IBM HTTP Server or Apache Server, edit the **httpd.conf** file on the Web server. Set the **AllowEncodedSlashes** directive to **on**; the directive should be added at the root level as a global directive.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>See the appropriate HTTP Server documentation</td>
<td>IBM HTTP Server</td>
</tr>
<tr>
<td>See the appropriate Apache Server documentation</td>
<td>AllowEncodedSlashes directives</td>
</tr>
</tbody>
</table>

4. Stop the Web server.

5. Install and configure the Web server plug-in on the system where the Web server is located using the Plug-ins installation wizard provided with WebSphere Application Server. Refer to Selecting a Web server topology diagram and roadmap for information. **If using WebDAV:** After successfully installing the Web server plug-in, locate and open your **plugin-cfg.xml** file and set **AcceptAllContent** to **true**.

**Important:** Depending on how you use the Web server, you may need to adjust the **ServerIOTimeout** value, which defines how long the plug-in should wait for a response from the application. The recommended minimum value is 60 but you may need to adjust this value higher if you are retrieving data from a database. To update this value, locate and open your **plugin-cfg.xml** file and set **ServerIOTimeout** to a value that is appropriate for your business needs. For additional information, see Common questions about the Web server plug-in.

6. If you are using Sun Web Server Version 7 update 8, read Technote 1448262 and perform the steps to resolve the HTTP 408/409 error.

7. Start the Web server.

**Parent topic:** Setting up a stand-alone server on i5/OS

**Previous topic:** Preparing DB2 for i5/OS for a stand-alone production server

**Next topic:** Configuring WebSphere Portal to use a user registry on IBM i5/OS
Configuring WebSphere Portal to use a user registry on IBM i5/OS

Configure user registry security on IBM® WebSphere® Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

Prior to configuring security, you should use the IBM WebSphere Application Server backupConfig task to create and store a backup of the IBM WebSphere Portal configuration; see backupConfig command for information.

**Prerequisites**
- Preparing your IBM i5/OS
- Installing WebSphere Portal on IBM i5/OS
- Preparing DB2 for i5/OS for a stand-alone production server

Perform the following tasks to configure WebSphere Portal to use a user registry:

1. **Preparing a Tivoli Directory Server for IBM i5/OS**
   If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

2. **Choosing your user registry model on IBM i5/OS**
   Choose between securing IBM WebSphere Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

3. **Adapting the attribute configuration**
   After installing IBM WebSphere Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

4. **Stand-alone server: Configuring WebSphere Portal to use dynamic groups on i5/OS**
   By default, WebSphere Portal is enabled for static groups. However, the Virtual Member Manager (VMM) allows users to be members of either static or dynamic groups. Static groups are those where a persistent binding exists between a group and its members. Dynamic groups are those where a search query is defined to retrieve the members of a group. If you have your LDAP server configured to use dynamic groups, complete the steps in this task for WebSphere Portal to use dynamic group queries when you setup your LDAP server.

5. **Stand-alone server: Enabling referrals for your LDAP user registry on i5/OS**
   Referrals redirect object requests from one LDAP server to another when objects do not exist or cannot be located in a particular directory tree. You should enable referrals if your environment has more than one user registry existing on multiple servers or domains.

**Related tasks**
Managing your user registry on i5/OS
Preparing a Tivoli Directory Server for IBM I5/OS

If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Tivoli Directory Server:

1. Customize the LDAP directory servers settings using the Directory Services Configuration Wizard. You must have *ALLOBJ and *IOSYSCFG special authority to use the wizard. Go to IBM System i and IBM i Information Center, select the appropriate Information Center version and navigate to e-business and Web serving > Security and IBM Tivoli Directory Server for I5/OS (LDAP) for information. Note: Due to a restriction in Tivoli Directory Server, users or groups must not contain a Turkish uppercase dotted I or lowercase dotted i in the DN as this will prevent correct retrieval of that user or group.

2. Perform the following steps to create the WebSphere Portal administrative user:

A. Optional: Perform the following steps to create a new directory suffix:

   1. Go to IBM System i and IBM i Information Center, select the appropriate Information Center version and navigate to Networking > TCP/IP applications, protocols, and services > IBM Directory Server for iSeries (LDAP) > Administering Directory Server > General administration tasks > Adding and Removing Directory Server suffixes for information.

   2. Stop and restart the LDAP server.

B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:

   - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
   - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.

C. Replace every dc=yourco,dc=com with your suffix.

D. Replace any prefixes and suffixes that are unique to your LDAP server.

E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.

F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.

G. Save your changes.

H. Follow the instructions provided with your directory server to import the LDIF file.

I. Stop and restart the LDAP server.

Parent topic: Configuring WebSphere Portal to use a user registry on IBM i5/OS

Next topic: Choosing your user registry model on IBM i5/OS
Choosing your user registry model on IBM i5/OS

Choose between securing IBM® WebSphere® Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

**- Prerequisites**
- Preparing a Tivoli Directory Server for IBM i5/OS

Choose one of the following user registry models:

**- Configuring a stand-alone LDAP user registry on IBM i5/OS**
Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

**- Configuring the default federated repository on IBM i5/OS**
Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

**Parent topic:** Configuring WebSphere Portal to use a user registry on IBM i5/OS

**Previous topic:** Preparing a Tivoli Directory Server for IBM i5/OS

**Next topic:** Adapting the attribute configuration
Configuring a stand-alone LDAP user registry on IBM i5/OS

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Configuring a stand-alone LDAP user registry on IBM i5/OS**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry to store all user account information for authorization.

- **Configuring a stand-alone LDAP user registry over SSL on IBM i5/OS**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

**Parent topic:** Choosing your user registry model on IBM i5/OS
Configuring a stand-alone LDAP user registry on IBM i5/OS

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry to store all user account information for authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps. If you need to rerun the wp-modify-ldap-security task to change the LDAP repositories or because the task failed, you must choose a new name for the realm using the standalone.ldap.realm parameter or you can set ignoreDuplicateIDs=true in the wkplc.properties file, before rerunning the task.

Perform the following steps to configure a standalone LDAP user registry:

Note: Use the wp_security_xxx.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_security_xxx.properties helper file.

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

2. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM Stand-alone LDAP configuration heading:

   - standalone.ldap.id
   - standalone.ldap.host
   - standalone.ldap.port
   - standalone.ldap.blindDN
   - standalone.ldap.bindPassword
   - standalone.ldap.idapServerType
   - standalone.ldap.userldMap
   - standalone.ldap.groupIdMap
   - standalone.ldap.groupMemberIdMap
   - standalone.ldap.userFilter
   - standalone.ldap.groupFilter
   - standalone.ldap.serverId
   - standalone.ldap.serverPassword
   - standalone.ldap.realm
   - standalone.ldap.primaryAdminId
   - standalone.ldap.primaryAdminPassword
   - standalone.ldap.primaryPortalAdminId
   - standalone.ldap.primaryPortalAdminPassword
   - standalone.ldap.primaryPortalAdminGroup
   - standalone.ldap.baseDN
3. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading:

   - `standalone.ldap.et.group.objectClasses`
   - `standalone.ldap.et.group.objectClassesForCreate`
   - `standalone.ldap.et.group.searchBases`
   - `standalone.ldap.et.personaccount.objectClasses`
   - `standalone.ldap.et.personaccount.objectClassesForCreate`
   - `standalone.ldap.et.personaccount.searchBases`

   **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

4. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attributes heading:

   - `standalone.ldap.gm.groupMemberName`
   - `standalone.ldap.gm.objectClass`
   - `standalone.ldap.gm.scope`
   - `standalone.ldap.gm.dummyMember`

   **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

5. Required: Enter a value for the following required relative distinguished name (RDN) parameters in the `wkplc.properties` file under the Default parent, RDN attribute heading:

   - `standalone.ldap.personAccountParent`
   - `standalone.ldap.groupParent`
   - `standalone.ldap.personAccountRdnProperties`
   - `standalone.ldap.groupRdnProperties`

6. Save your changes to the `wkplc.properties` file.

7. Optional: Enter the following Web content authors parameters in the `wkplc_comp.properties` file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product.

   - `WcmContentAuthorsGroupId`
   - `WcmContentAuthorsGroupCN`

   **Note:** See the `wkplc_comp.properties` file for specific information about the required parameters and for advanced parameters.

8. Run the `ConfigEngine.sh validate-standalone-ldap -DWasPassword=password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, WasPassword is the value entered during installation and not a value found in your LDAP user registry.

   **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press `y` then `Enter`.

9. Run the `ConfigEngine.sh wp-modify-ldap-security -DWasPassword=password` task, from the `wp_profile_root` /ConfigEngine directory, to set the standalone LDAP user registry.

10. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

11. Run the `ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root` /ConfigEngine directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

12. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the
LDAP directory.

**Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

A. Edit the `wp_profile_root`/

   `/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties` file.

B. Add the following lines to the file: `uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN` `cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN` Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: **Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management**

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

13. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

**Parent topic:** Configuring a stand-alone LDAP user registry on IBM i5/OS

**Related tasks**

- Adapting the attribute configuration
- Using the member fixer tool with IBM Lotus Web Content Management
- Starting and stopping servers, deployment managers, and node agents

---

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type of LDAP</strong></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>
Configuring a stand-alone LDAP user registry over SSL on IBM i5/OS

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to configure a standalone LDAP user registry over SSL:

Note: Use the wp_security_xxx.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_security_xxx.properties helper file.

1. To specify the LDAP server's SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:
   - Server trust store
     A. Add the certificate to the trust store:
        1. Log in to the WebSphere Application Server Administrative Console.
        2. Navigate to Security > SSL certificate and key management > SSL configurations.
        3. Click the appropriate SSL configuration from the list. For example, Clustered environments: CellDefaultSSLSettings
           - Clustered environments: CellDefaultSSLSettings
        4. Click Key stores and certificates.
        5. Click the appropriate trust store from the list; for example, NodeDefaultTrustStore.
        6. Click Signer certificates, click Add, and then enter the following information:
           - Type the Alias the key store uses for the signer certificate.
           - Type the File name where the signer certificate is located.
        7. Click OK and then click Save to save the changes to the master configuration.
   B. Retrieve the certificate from the port:
      1. Log in to the WebSphere Application Server Administrative Console.
      2. Navigate to Security > SSL certificate and key management > SSL configurations.
      3. Click the appropriate SSL configuration from the list. For example, NodeDefaultSSLSettings

Clustered environments: **CellDefaultSSLSettings**

**Clustered environments:** Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.

4. Click **Key stores and certificates**.

5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.

6. Click **Signer certificates**, click **Retrieve from port**, and then enter the following information:
   - Type the **Host** name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the SSL **Port** used when attempting to retrieve the signer certificate.
   - Type the **Alias** the key store uses for the signer certificate.

7. Click **Retrieve signer information** to retrieve the certificate from the port.

8. Click **OK** and then click **Save** to save the changes to the master configuration.

**Client trust store**

- **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

A. See **Secure installation for client signer retrieval**.

B. Run the **retrieveSigners** task from the **wp_profile_root/bin** directory; see **retrieveSigners command** for information. In a deployed environment, you will need to run the retrieveSigners task, for any federated node, against the Deployment Manager. **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

C. Update the trust store properties file:

1. Use a text editor to open the **ssl.client.props** file, located in the **wp_profile_root/properties** directory.

2. Change the **com.ibm.ssl.trustStore** parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter **com.ibm.ssl.trustStore=${CONFIG_ROOT}/cells/wpsbvt/nodes/wpsbvt/trust.p12** to use the default trust store.

3. Save your changes.

2. Use a text editor to open the **wkplc.properties** file, located in the **wp_profile_root/ConfigEngine/properties** directory.

3. Required: Enter a value for the following required parameters in the **wkplc.properties** file under the **VMM Standalone LDAP configuration** heading. **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - **standalone.idap/idp
d - standalone.idap/host
d - standalone.idap/port
d - standalone.idap.blindDN
d - standalone.idap.bindPassword
d - standalone.idap.idapServerType
d - standalone.idap.userldMap
d - standalone.idap.groupldMap
d - standalone.idap.groupMemberIdMap
d - standalone.idap.userFilter
d - standalone.idap.groupFilter
d - standalone.idap.serverld
d - standalone.idap.serverPassword
d - standalone.idap.realm
d - standalone.idap.primaryAdminId
d - standalone.idap.primaryAdminPassword
d - standalone.idap.primaryPortalAdminId
4. Required: Enter a value for the following required entity types parameters in the wkplc.properties file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.et.group.objectClasses
   - standalone.ldap.et.group.objectClassesForCreate
   - standalone.ldap.et.group.searchBases
   - standalone.ldap.et.personaccount.objectClasses
   - standalone.ldap.et.personaccount.objectClassesForCreate
   - standalone.ldap.et.personaccount.searchBases

5. Required: Enter a value for the following required group member parameters in the wkplc.properties file under the Group member attributes heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.gm.groupMemberName
   - standalone.ldap.gm.objectClass
   - standalone.ldap.gm.scope
   - standalone.ldap.gm.dummyMember

6. Required: Enter a value for the following required relative distinguished name (RDN®) parameters in the wkplc.properties file under the Default parent, RDN attribute heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.personAccountParent
   - standalone.ldap.groupParent
   - standalone.ldap.personAccountRdnProperties
   - standalone.ldap.groupRdnProperties

7. Enter a value for the following parameters to enable Secure Socket Layers (SSL): **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   Required parameters:
   - standalone.ldap.sslEnabled
   - standalone.ldap.sslConfiguration

   Optional parameters:
   - standalone.ldap.certificateMapMode
   - standalone.ldap.certificateFilter

8. Save your changes to the wkplc.properties file.

9. Run the `ConfigEngine.sh validate-standalone-ldap -DWasPassword=password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, `WasPassword` is the value entered during installation and not a value found in your LDAP user registry. **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now? Press `y` then `Enter`.

10. Run the `ConfigEngine.sh wp-modify-ldap-security -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to set the standalone LDAP user registry.

11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: **Starting and stopping servers, deployment managers, and node agents.**

12. Run the `ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured...
LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

**Parent topic:** Configuring a stand-alone LDAP user registry on IBM i5/OS

**Related tasks**
Adapting the attribute configuration
Starting and stopping servers, deployment managers, and node agents
Configuring the default federated repository on IBM i5/OS

Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

Perform the following tasks to add user registries to the default federated repository:

- **Configuring a federated LDAP user registry on IBM i5/OS**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Adding a database user registry on IBM i5/OS**
  Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

- **Adding realm support on IBM i5/OS**
  A realm is a group of users from one or more user registries that form a coherent group within IBM WebSphere Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

**Parent topic:** Choosing your user registry model on IBM i5/OS
Configuring a federated LDAP user registry on IBM i5/OS

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal. Choose one of the following options to configure your LDAP server:

- **Adding an LDAP user registry on IBM i5/OS**
  Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time. If IBM Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=groupName` and the hierarchical format is `cn=groupName, o=root`. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as `wpsadmin`, this ID cannot exist in the LDAP you are adding.

- **Adding an LDAP user registry over SSL on IBM i5/OS**
  Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

**Parent topic:** Configuring the default federated repository on IBM i5/OS
Adding an LDAP user registry on IBM i5/OS

Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

If IBM® Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=`groupName and the hierarchical format is `cn=`groupName,o=root. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as wpsadmin, this ID cannot exist in the LDAP you are adding.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

1. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

2. Required: Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated LDAP Properties heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.

   - `federated.ldap.id`
   - `federated.ldap.host`
   - `federated.ldap.port`
   - `federated.ldap.bindDN`
   - `federated.ldap.bindPassword`
   - `federated.ldap.ldapServerType`
   - `federated.ldap.baseDN`

3. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.

   - `federated.ldap.et.group.objectClasses`
   - `federated.ldap.et.group.objectClassesForCreate`
   - `federated.ldap.et.group.searchBases`
   - `federated.ldap.et.personaccount.objectClasses`
   - `federated.ldap.et.personaccount.objectClassesForCreate`
   - `federated.ldap.et.personaccount.searchBases`
4. Required: Enter a value for the following required group member parameters in the wkplc.properties file under the Group member attribute heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - federated.ldap.gm.groupMemberName
   - federated.ldap.gm.objectClass
   - federated.ldap.gm.scope
   - federated.ldap.gm.dummyMember

5. Save your changes to the wkplc.properties file.

6. Optional: Enter the following Web content authors parameters in the wkplc_comp.properties file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. Note: See the wkplc_comp.properties file for specific information about the required parameters and for advanced parameters.
   - WcmContentAuthorsGroupId
   - WcmContentAuthorsGroupCN

7. Run the ConfigEngine.sh validate-federated-ldap -DWasPassword=password task to validate your LDAP server settings. Attention: If you have not deleted the default file repository, WasPassword is the value entered during installation and not a value found in your LDAP user registry.
   Note: During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press y then Enter.

8. Run the ConfigEngine.sh wp-create-ldap -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add an LDAP user registry to the default federated repository. Note: Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

10. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:
    A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
    B. Enter a value for the following required parameters in the wkplc.properties file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms:
       Note: See the properties file for specific information about the required parameters and for advanced parameters.
       - id
       - baseDN
       - nameInRepository
    C. Save your changes to the wkplc.properties file.
    D. Run the ConfigEngine.sh wp-create-base-entry -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to create a base entry in a repository.
    E. Stop and restart all necessary servers to propagate your changes.

11. Optional: Run the ConfigEngine.sh wp-query-repository -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to list the names and types of configured repositories.

12. Run the ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to check that all defined attributes are available in the configured LDAP user registry. Important: When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between
WebSphere Portal and the LDAP server.

13. Perform the following steps to update the user registry where new users and groups are stored:

   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - `personAccountParent`
      - `groupParent`
      - `personAccountRdnProperties`
      - `groupRdnProperties`

      The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:

      - `personAccountParent=dc=yourco,dc=com`
      - `groupParent=dc=yourco,dc=com`

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root` directory, to delete the old attributes before adding the new attributes.

   E. Stop and restart all necessary servers to propagate your changes.

14. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for `realmName` or leave blank to update the default realm.

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

   E. Stop and restart all necessary servers to propagate your changes.

15. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the `contentAuthors` group is correctly mapped to the appropriate group in the LDAP directory.

   **Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

   A. Edit the `wp_profile_root/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties` file.

   B. Add the following lines to the file:

   ```
   uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
   cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
   
   Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.
   ```
C. Save your changes and close the file.
D. Run the `ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone ldap realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated realm</code> is empty, use <code>default WIM File Based Realm</code> as the default value.</td>
</tr>
</tbody>
</table>

16. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

17. Optional: **This step is required in a production environment.** Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

   **Important:**
   - Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
   - Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

   **Note:** If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

   A. Run the following task from the `wp_profile_root/ConfigEngine` directory: `ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword` **Important:** You must provide the full distinguished name (DN) for the `newAdminid` parameter.

   B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

   C. Run the following task from the `wp_profile_root/ConfigEngine` directory: `ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadminGroupId` **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

   D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

18. Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it. The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by
duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: *Deleting the repository.*

**Parent topic:** Configuring a federated LDAP user registry on IBM i5/OS

**Related tasks**
Adapting the attribute configuration  
Deleting the repository on i5/OS  
Starting and stopping servers, deployment managers, and node agents

**Related information**
User IDs and passwords
Adding an LDAP user registry over SSL on IBM i5/OS

Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry over SSL to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

**Note:** Use the wp_add_federated_xxx.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_add_federated_xxx.properties helper file.

1. To specify the LDAP server's SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:

   - **Server trust store**
     - A. Add the certificate to the trust store:
       1. Log in to the WebSphere Application Server Administrative Console.
       2. Navigate to Security > SSL certificate and key management > SSL configurations.
       3. Click the appropriate SSL configuration from the list. For example, - Stand-alone environments: NodeDefaultSSLSettings
          - Clustered environments: CellDefaultSSLSettings

       **Clustered environments:** Ensure the setting for SSL configuration for outbound connection matches your SSL settings.

       4. Click Key stores and certificates.
       5. Click the appropriate trust store from the list; for example, NodeDefaultTrustStore.

       6. Click Signer certificates, click Add, and then enter the following information:

          - Type the Alias the key store uses for the signer certificate.
          - Type the File name where the signer certificate is located.

       7. Click OK and then click Save to save the changes to the master configuration.

   - B. Retrieve the certificate from the port:
     1. Log in to the WebSphere Application Server Administrative Console.
     2. Navigate to Security > SSL certificate and key management > SSL configurations.
     3. Click the appropriate SSL configuration from the list. For example,

       - Stand-alone environments: NodeDefaultSSLSettings
Clustered environments: CellDefaultSSLSettings

Clustered environments: Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.

4. Click **Key stores and certificates**.

5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.

6. Click **Signer certificates**, click **Retrieve from port**, and then enter the following information:
   - Type the **Host** name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the **SSL Port** used when attempting to retrieve the signer certificate.
   - Type the **Alias** the key store uses for the signer certificate.

7. Click **Retrieve signer information** to retrieve the certificate from the port.

8. Click **OK** and then click **Save** to save the changes to the master configuration.

**Client trust store**

- **Note**: This task might report an error, but it does successfully update the trust store. You can ignore the error message.

A. See **Secure installation for client signer retrieval**.

B. Run the **retrieveSigners** task from the **wp_profile_root\bin** directory; see **retrieveSigners command for information**. In a deployed environment, you will need to **run the retrieveSigners task, for any federated node, against the Deployment Manager**. **Note**: This task might report an error, but it does successfully update the trust store. You can ignore the error message.

C. Update the trust store properties file:

1. Use a text editor to open the **ssl.client.props** file, located in the **wp_profile_root\properties** directory.

2. Change the **com.ibm.ssl.trustStore** parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter **com.ibm.ssl.trustStore=${CONFIG_ROOT}\cells\wpsbvt\nodes\wpsbvt\trust.p12** to use the default trust store.

3. Save your changes.

2. Use a text editor to open the **wkplc.properties** file, located in the **wp_profile_root\ConfigEngine\properties** directory.

3. Required: Enter a value for the following required parameters in the **wkplc.properties** file under the VMM Federated LDAP Properties heading: **Note**: See the properties file for specific information about the required parameters and for advanced parameters.

   - **federated.ldap.id**
   - **federated.ldap.host**
   - **federated.ldap.port**
   - **federated.ldap.bindDN**
   - **federated.ldap.bindPassword**
   - **federated.ldap.ldapServerType**
   - **federated.ldap.baseDN**

4. Required: Enter a value for the following required entity types parameters in the **wkplc.properties** file under the LDAP entity types heading: **Note**: See the properties file for specific information about the required parameters and for advanced parameters.

   - **federated.ldap.et.group.objectClasses**
   - **federated.ldap.et.group.objectClassesForCreate**
   - **federated.ldap.et.group.searchBases**
   - **federated.ldap.et.personaccount.objectClasses**
   - **federated.ldap.et.personaccount.objectClassesForCreate**
   - **federated.ldap.et.personaccount.searchBases**
5. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attribute heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `federated.ldap.gm.groupMemberName`
   - `federated.ldap.gm.objectClass`
   - `federated.ldap.gm.scope`
   - `federated.ldap.gm.dummyMember`

6. Enter a value for the following parameters to enable Secure Socket Layers (SSL): **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   Required parameters:
   - `federated.ldap.sslEnabled`
   - `federated.ldap.sslConfiguration`

   Optional parameters:
   - `federated.ldap.certificateMapMode`
   - `federated.ldap.certificateFilter`

7. Save your changes to the `wkplc.properties` file.

8. Optional: Enter the following Web content authors parameters in the `wkplc_comp.properties` file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. **Note:** See the `wkplc_comp.properties` file for specific information about the required parameters and for advanced parameters.
   - `WcmContentAuthorsGroupId`
   - `WcmContentAuthorsGroupCN`

9. Run the `ConfigEngine.sh validate-federated-ldap -DWasPassword=password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, `WasPassword` is the value entered during installation and not a value found in your LDAP user registry. **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press **y** then **Enter**.

10. Run the `ConfigEngine.sh wp-create-ldap -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to add an LDAP user registry to the default federated repository. **Note:** Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents*.

12. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:
   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
      - `id`
      - `baseDN`
- **nameInRepository**

C. Save your changes to the wkplc.properties file.

D. Run the `ConfigEngine.sh wp-create-base-entry -DWasPassword=password` task, from the `wp_profile_root`/ConfigEngine directory, to create a base entry in a repository.

E. Stop and restart all necessary servers to propagate your changes.

13. Optional: Run the `ConfigEngine.sh wp-query-repository -DWasPassword=password` task, from the `wp_profile_root`/ConfigEngine directory, to list the names and types of configured repositories.

14. Run the `ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root`/ConfigEngine directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

15. Perform the following steps to update the user registry where new users and groups are stored:

   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root`/ConfigEngine/properties directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - `personAccountParent`
      - `groupParent`
      - `personAccountRdnProperties`
      - `groupRdnProperties`

      The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:

      - `personAccountParent=dc=yourco,dc=com`
      - `groupParent=dc=yourco,dc=com`

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root`/ConfigEngine directory, to delete the old attributes before adding the new attributes.

   E. Stop and restart all necessary servers to propagate your changes.

16. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root`/ConfigEngine/properties directory.

   B. Enter a value for `realmName` or leave blank to update the default realm.

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root`/ConfigEngine directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

   E. Stop and restart all necessary servers to propagate your changes.

17. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the
Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

**Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

A. Edit the `wp_profile_root` file:

```
/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties
```

B. Add the following lines to the file:

```
uid=xyzadmin, o=defaultWIMFileBasedRealm -> portal_admin_DN

cn=contentauthors, o=defaultWIMFileBasedRealm -> content_authors_group_DN
```

Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

18. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

19. Optional: **This step is required in a production environment.** Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

   **Important:**
   - Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
   - Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

   **Note:** If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

A. Run the following task from the `wp_profile_root` directory:

```
ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword
```

   **Important:** You must provide the full distinguished name (DN) for the `newAdminid` parameter.

B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.
C. Run the following task from the `wp_profile_root/ConfigEngine` directory:

```
ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid
```

**Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

20. Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it. The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: *Deleting the repository.*

**Parent topic:** Configuring a federated LDAP user registry on IBM i5/OS

**Related tasks**

Adapting the attribute configuration
Deleting the repository on i5/OS
Starting and stopping servers, deployment managers, and node agents

**Related information**

User IDs and passwords
Adding a database user registry on IBM i5/OS

Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add a database user registry to the default federated repository; you must repeat these steps for each additional database user registry that you plan to add:

**Note:** Use the `wp_add_DB.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_add_DB.properties` helper file.

---

Perform the following steps to create the DB2 for i5/OS database:

**Instructions for setting up databases:** Refer to the appropriate documentation for the type of database you want to set up.

**Consulting your database administrator:** The task of setting up a new database is typically performed by a database administrator. However, the following steps are provided for your reference in the event you are creating a stand-alone database for testing or demonstration purposes. Consult your database administrator before proceeding with the following steps if you plan to create a database for a production environment.

- **A.** Login to a remote i5/OS session.
- **B.** Enter the `strsql` command to start the interactive sql session.
- **C.** Enter the `create schema database_name` command, where `database_name` is the name you want to use for the database.

Perform the following steps to define the `DbDriver` and `DbLibrary` parameter values:

- **A.** Navigate to the following directory: `wp_profile_root/ConfigEngine/properties` directory.
- **B.** Locate and open `wkplc_dbtype.properties` with any text editor.
- **C.** Enter a value for the following parameters under the appropriate database type properties heading:
  - `db_type.DbDriver`
  - `db_type.DbLibrary`
- **D.** Save your changes.

Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated Database Properties heading:

**Note:** See the properties file for specific information about the required parameters and for advanced parameters.

- `federated.db.DataSourceName`
- `federated.dbDbType`
- `federated.db.DbUrl`
- `federated.db.Id`
- federated.db.baseDN
- federated.db.DbUser
- federated.db.DbPassword
- federated.db.DbName

5. Change the value for the **com.ibm.SOAP.requestTimeout** parameter to 1000.
   A. Navigate to the following directory: **wp_profile_root/properties**
   B. Locate and open soap.client.props with any text editor.
   C. Locate the **com.ibm.SOAP.requestTimeout** parameter and change the value to 1000.
   D. Save and close soap.client.props.

6. Perform the following steps in a clustered environment:
   A. Run the **ConfigEngine.sh wp-prep-vmm-db-secured-environment** task from the **wp_profile_root/ConfigEngine** directory to create the local Deployment Manager WebSphere variable used to access the database jars. **Note:** The **db_type** in **db_type.DmgrDbLibrary** should be set to the type of database you are using, for example db2_iseries. The **local path of the database jars on the Deployment Manager** should be one of the following options:
      - **DB2 for i5/OS Type 2 driver:** /QIBM/ProdData/Java400/ext/db2_classes.jar
      - **DB2 for i5/OS Type 4 driver:** /QIBM/ProdData/HTTP/Public/jt400/lib/jt400.jar
   B. Run the following task. Include each node name as a comma separated list in the command:
      **Running the task:** You do not have to run this task more than once. You can run this task from any node in the cluster.
      1. Set the property value for **federated.db.DbType** if using a database user registry and set the property value for **laDbType** if using a property extension database in the wpkc.properties file.
      2. Run the **ConfigEngine.sh wp-node-prep-vmm-db-secured-environment** task from the **wp_profile_root/ConfigEngine** directory on each node to create the variable used to access the VMM database jars. **Note:** **VmmNodeName** is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The **db_type in db_type.NodeDbLibrary** should be set to the type of database you are using, for example db2.
      - **DB2 for i5/OS Type 2 driver:** /QIBM/ProdData/Java400/ext/db2_classes.jar
      - **DB2 for i5/OS Type 4 driver:** /QIBM/ProdData/HTTP/Public/jt400/lib/jt400.jar
   C. Stop and restart all necessary servers to propagate your changes.
   7. Run the **ConfigEngine.sh wp-create-db** task, from the **wp_profile_root/ConfigEngine** directory, to add a database user registry to the default federated repository. **Note:** Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.
   8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: **Starting and stopping servers, deployment managers, and node agents.**
   9. Perform the following steps to update the user registry where new users and groups are stored:
      **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.
Attention: During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:
   
   **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   
   - `personAccountParent`
   - `groupParent`
   - `personAccountRdnProperties`
   - `groupRdnProperties`

   The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:
   
   - `personAccountParent=dc=yourco,dc=com`
   - `groupParent=dc=yourco,dc=com`

C. Save your changes to the `wkplc.properties` file.

D. Run the `ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to delete the old attributes before adding the new attributes.

E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Run the `ConfigEngine.sh wp-query-repository -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to list the names and types of configured repositories.

**Parent topic:** Configuring the default federated repository on IBM i5/OS

**Parent topic:** Updating your user registry on i5/OS
Adding realm support on IBM i5/OS

A realm is a group of users from one or more user registries that form a coherent group within IBM® WebSphere® Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

Before configuring realm support, you must add all LDAP user registries and/or database user registries, that you will use to create a single realm or multiple realms, to the federated repository. If you are going to create multiple realms, you must create all required base entries within your LDAP user registries and/or database user registries. All base entry names must be unique within the federated repository. In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add realm support to your user registry model:

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
   - Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading:
     - realmName
     - securityUse
     - delimiter
     - addBaseEntry
   - Note: See the properties file for specific information about the required parameters and for advanced parameters.

2. Save your changes to the wkplc.properties file.

3. Run the ConfigEngine.sh wp-create-realm -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add a new realm to the Virtual Member Manager configuration. Important: To create multiple realms, ensure that your federated repository contains the required unique base entries. Stop and restart the appropriate servers for your installation environment, and then update the wkplc.properties file with the base entry information and rerun the wp-create-realm task. Repeat these steps until all realms are created.

4. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

5. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading and then save your changes:
   - realmName
   - realm.personAccountParent
   - realm.groupParent
   - realm.orgContainerParent

6. Run the ConfigEngine.sh wp-modify-realm-defaultparents -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to update the default parents per entity type and realm. Important: Stop and restart the appropriate servers for your installation environment before rerunning this task for any additional entity types and realms.
8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

9. Optional: Perform the following steps to add additional base entries to the realm configuration; for example, if you had two additional base entries (base entry 1 and base entry 2) to add to the realm you just created, you would update the wkplc.properties file with the information from base entry 1 and then run this task. Then you would update the properties file with the information for base entry 2 and then run this task:
   A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
   B. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading:
      - realmName
      - addBaseEntry
   C. Save your changes to the wkplc.properties file.
   D. Run the ConfigEngine.sh wp-add-realm-baseentry -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add additional LDAP base entries to the realm configuration.
   E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID; this step is required if you change the default realm:
    A. Create a new user in the Manage Users and Groups portlet to replace the current WebSphere Application Server administrative user.
    B. Create a new user in the Manage Users and Groups portlet to replace the current WebSphere Portal administrative user.
    C. Create a new group in the Manage Users and Groups portlet to replace the current group.
    D. Run the ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid task, from the wp_profile_root/ConfigEngine directory, to replace the old WebSphere Application Server administrative user ID and group ID with the new user and group. **Important:** You must provide the full distinguished name (DN) for the newAdminId and newAdminGroupId parameters.
    Additional parameter for stopped servers: This task verifies the user against a running server instance. If the server is stopped, add the -Dskip.ldap.validation=true parameter to the task to skip the validation.
    E. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.
    F. Run the ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid task to replace the old WebSphere Portal administrative user ID and group ID with the new user and group. **Important:** You must provide the full distinguished name (DN) for the newAdminId and newAdminGroupId parameters.
    Additional parameter for stopped servers: This task verifies the user against a running server instance. If the server is stopped, add the -Dskip.ldap.validation=true parameter to the task to skip the validation.
    G. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

11. Optional: Perform the following steps to set the realm you created as the default realm: **Remember:** Only users defined in base entries that exist in the default realm are able to log into WebSphere Portal. If you find that a user cannot log in to WebSphere Portal, check to see if the base entry that contains the user exists in the default realm. You can run the wp-query-realm-baseentry task to see what base entries are part of the default realm. If the default realm is missing the base entry, run the wp-add-realm-baseentry task to add the base entry to the default realm.
A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. For `defaultRealmName`, type the `realmName` property value you want to use as the default realm.

C. Save your changes to the `wkplc.properties` file.

D. Run the `ConfigEngine.sh wp-default-realm -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to set this realm as the default realm.

E. Stop and restart all necessary servers to propagate your changes.

**12. Optional:** Perform the following steps to query a realm for a list of its base entries:

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. For `realmName`, type the name of the realm you want to query.

C. Save your changes to the `wkplc.properties` file.

D. Run the `ConfigEngine.sh wp-query-realm-baseentry -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to list the base entries for a specific realm.

**13. Optional:** Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. Enter a value for `realmName` or leave blank to update the default realm.

C. Save your changes to the `wkplc.properties` file.

D. Run the `ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

E. Stop and restart all necessary servers to propagate your changes.

**Parent topic:** Configuring the default federated repository on IBM i5/OS
Adapting the attribute configuration on IBM i

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

### Prerequisites

- Preparing a Tivoli Directory Server for IBM i5/OS
- Choosing your user registry model on IBM i5/OS

After installation, IBM WebSphere Portal has a predefined set of attributes for users and groups. Your LDAP server may have a different set of predefined user and group attributes. To ensure proper communication between WebSphere Portal and your LDAP server, you can configure additional attributes and flag existing attributes as required or unsupported on a per repository basis or for all configure repositories.

LDAP servers can only handle attributes that are explicitly defined in their schema. The LDAP server's schema is different from the WebSphere Portal schema but the two schemas should match for proper communication between WebSphere Portal and the LDAP server. The task to add the LDAP user registry does some basic attribute configurations depending on the type of LDAP server that you choose. You may, however, still need to adapt the WebSphere Portal configuration to match the LDAP schema; for example, if an attribute is defined in WebSphere Portal but not in the LDAP server, you will need to perform one of the following tasks to resolve this mismatch:

- Flag the attribute as unsupported for the LDAP server
- Introduce an attribute mapping that maps the WebSphere Portal attribute to an attribute defined in the LDAP schema

Perform the following tasks to adapt the attribute configuration to match the configured LDAP server(s) and your business needs:

1. **Querying the defined attributes on IBM i5/OS**
   
   After installing IBM WebSphere Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

2. **Adding attributes on IBM i5/OS**
   
   The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM WebSphere Portal and your user registry.

3. **Mapping attributes on IBM i5/OS**
   
   After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

4. **Removing attributes**
   
   Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

**Parent topic:** Configuring WebSphere Portal to use a user registry on IBM i5/OS

**Previous topic:** Choosing your user registry model on IBM i5/OS

**Next topic:** Stand-alone server: Configuring WebSphere Portal to use dynamic groups on i5/OS
Related tasks
Adding an LDAP user registry on IBM i5/OS
Adding an LDAP user registry over SSL on IBM i5/OS
Configuring a stand-alone LDAP user registry on IBM i5/OS
Configuring a stand-alone LDAP user registry over SSL on IBM i5/OS
Querying the defined attributes on IBM i5/OS

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

**Warning:** If you are using a database user registry or a property extension database, copy the database drivers to the `AppServer_root/lib` directory before executing this script.

Run the `ConfigEngine.sh wp-query-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, any time during the configuration process or at runtime to query an overview of the currently defined attributes. This task creates the `availableAttributes.html` report, located in the `wp_profile_root/ConfigEngine/log` directory. The report contains one table that lists the available attributes for Users (**PersonAccount**) and one table that lists the available attributes for Groups. For each configured repository there is a column that indicates if the attribute is flagged as unsupported or if the attribute is mapped to a different LDAP attribute. **Note:** This task does not validate the existence of attributes in the LDAP schema.

**Parent topic:** Adapting the attribute configuration

**Next topic:** Adding attributes on IBM i5/OS
Adding attributes on IBM i5/OS

The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM® WebSphere® Portal and your user registry.

- Prerequisites
  - Querying the defined attributes on IBM i5/OS

Perform the following steps to add new attributes to your user registry:

1. Install the required Enterprise Archive (.ear) file on WebSphere Application Server. Select the appropriate task to install the .ear file:

   **Table 1. Steps for installing the .ear file by environment**

<table>
<thead>
<tr>
<th>Environment</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stand-alone environment</strong></td>
<td>Perform the following steps to install the .ear file in a stand-alone environment: Open a command prompt. Navigate to the <code>wp_profile_root/ConfigEngine</code> directory. Run the <code>ConfigEngine.sh wp-la-install-ear -DWasPassword=password</code> task.</td>
</tr>
<tr>
<td><strong>Clustered environment</strong></td>
<td>Perform the following steps to install the .ear file in a clustered environment: Open a command prompt. Navigate to the <code>wp_profile_root/ConfigEngine</code> directory on the primary node. Run the <code>ConfigEngine.sh wp-la-install-ear -DWasPassword=dmgr_password -DServerName=dmgr -DNodeName=node_name</code> task. Where <code>node_name</code> is the name of the node where the deployment manager resides; you can find the <code>node_name</code> value in the WebSphere Application Server Administrative Console under <strong>System administrator &gt; Deployment Manager &gt; Runtime tab &gt; General Properties &gt; Node Name</strong>.</td>
</tr>
</tbody>
</table>

2. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: **Starting and stopping servers, deployment managers, and node agents**.

3. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

4. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Property Extension Properties heading: **Note**: See the properties file for specific information about the required parameters and for advanced parameters.
   - `la.providerURL`
   - `la.propertyName`
   - `la.entityTypes`
   - `la.dataType`
   - `la.multiValued`
5. Save your changes to the wkplc.properties file.
6. Run the `ConfigEngine.sh wp-add-property -DWasPassword=password` task to add the attribute to the user registry.

   **Note**: This task performs an EJB call to WebSphere Application Server, which must authenticate against WebSphere Application Server. Depending on the configuration in the sas.client.props file, you may receive a popup window or a command line prompt asking for user identity and password. Enter the WebSphere Application Server user ID and password.

   **Remember**: If you have multiple properties to add, repeat all steps, except for the `wp-la-install-ear` task, until all new attributes are added.

7. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents.*

**Parent topic:** Adapting the attribute configuration  
**Previous topic:** Querying the defined attributes on IBM i5/OS  
**Next topic:** Mapping attributes on IBM i5/OS  

**Related tasks**
Starting and stopping servers, deployment managers, and node agents
Mapping attributes on IBM i5/OS

After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

**Prerequisites**
- Querying the defined attributes on IBM i5/OS
- Adding attributes on IBM i5/OS

Perform the following steps to map attributes between WebSphere Portal and your LDAP server; if you have multiple LDAP servers, you will need to perform these steps for each LDAP server:

1. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
2. Enter a value for one of the following sets of parameters in the `wkplc.properties` file to identify your LDAP server:
   - **Note:** Make sure you use the same values you used to configure your LDAP server.

<p>| Table 1. Identifying your LDAP server in the wkplc.properties file. |</p>
<table>
<thead>
<tr>
<th>Repository type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stand-alone</strong></td>
<td>The following parameters are found under the LDAP attribute configuration heading: <strong>Note:</strong> See the properties file for specific information about the required parameters and for advanced parameters.</td>
</tr>
<tr>
<td></td>
<td><code>standalone.ldap.id</code> <code>standalone.ldap.host</code> <code>standalone.ldap.port</code> <code>standalone.ldap.sslEnabled</code> <code>standalone.ldap.bindDN</code> <code>standalone.ldap.bindPassword</code> <code>standalone.ldap.baseDN</code></td>
</tr>
<tr>
<td><strong>Federated</strong></td>
<td>The following parameters are found under the VMM Federated repository properties heading: <strong>Note:</strong> See the properties file for specific information about the required parameters and for advanced parameters.</td>
</tr>
<tr>
<td></td>
<td><code>federated.ldap.id</code> <code>federated.ldap.host</code> <code>federated.ldap.port</code> <code>federated.ldap.sslEnabled</code> <code>federated.ldap.bindDN</code> <code>federated.ldap.bindPassword</code></td>
</tr>
</tbody>
</table>

3. Run one of the following tasks to check that all defined attributes are available in the configured LDAP user registry:

<p>| Table 2. Task to check that all defined attributes are available in the configured LDAP user registry. |</p>
<table>
<thead>
<tr>
<th>Repository type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stand-alone</strong></td>
<td><code>ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory.</code></td>
</tr>
<tr>
<td><strong>Federated</strong></td>
<td><code>ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory.</code></td>
</tr>
</tbody>
</table>
4. Open the ConfigTrace.log file, located in the `wp_profile_root/ConfigEngine/log` directory, to review the following output for the **PersonAccount** and **Group** entity type:

   - **The following attributes are defined in WebSphere Portal but not in the LDAP server**
     - This list contains all attributes that are defined in WebSphere Portal but not available in the LDAP. Flag attributes that you do not plan to use in WebSphere Portal as unsupported. Map the attributes that you plan to use to the attributes that exist in the LDAP; you must also map the `uid, cn, firstName, sn, preferredLanguage, and ibm-primaryEmail` attributes if they are contained in the list.

   - **The following attributes are flagged as required in the LDAP server but not in WebSphere Portal**
     - This list contains all attributes that are defined as "MUST" in the LDAP server but not as required in WebSphere Portal. You should flag these attributes as required within WebSphere Portal; see the step below about flagging an attribute as either unsupported or required.

   - **The following attributes have a different type in WebSphere Portal and in the LDAP server**
     - This list contains all attributes that WebSphere Portal might ignore because the data type within WebSphere Portal and within the LDAP server do not match.

5. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

6. Enter a value for one of the following sets of parameters in the `wkplc.properties` file to correct any issues found in the config trace file: *Table 3. Parameters to define in the wkplc.properties file to correct any issues found in the config trace file.*

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Parameters</th>
</tr>
</thead>
</table>
| Stand-alone     | The following parameters are found under the LDAP attribute configuration heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters. 
  - `standalone.ldap.idstandalone.ldap.attributes.nonSupported=standalone.ldap.attributes.nonSupported.delete=standalone.ldap.attributes.mapping.ldapName=standalone.ldap.attributes.mapping.portalName=standalone.ldap.attributes.mapping.entityTypes=
  - `standalone.ldap.attributes.nonSupported.delete=standalone.ldap.attributes.mapping.ldapName=standalone.ldap.attributes.mapping.portalName=standalone.ldap.attributes.mapping.entityTypes=
  - `ibm-primaryEmail=ibm-jobTitle` for both the **PersonAccount** and **Group** entity types. |

613
7. Save your changes to the wkplc.properties file.

8. Run one of the following tasks to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between WebSphere Portal and the LDAP user registry:

   Table 4. Task to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between Portal and the LDAP user registry.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>ConfigEngine.sh wp-update-standalone-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory</td>
</tr>
<tr>
<td>Federated</td>
<td>ConfigEngine.sh wp-update-federated-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory</td>
</tr>
</tbody>
</table>

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

10. Optional: Perform the following steps to flag an attribute as either unsupported or required for the entire WebSphere Portal environment instead of just for the specified LDAP:

   A. Enter a value for the following required parameters in the wkplc.properties file:

      - **user.attributes.required**
      - **user.attributes.nonsupported**

   B. Save your changes to the wkplc.properties file.

   C. Run the ConfigEngine.sh wp-update-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory.

   D. Stop and restart all necessary servers to propagate your changes.

Parent topic: Adapting the attribute configuration

Previous topic: Adding attributes on IBM i5/OS
Next topic: Removing attributes

Related tasks
Starting and stopping servers, deployment managers, and node agents
Removing attributes

Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

- Prerequisites
  - Querying the defined attributes on IBM i5/OS
  - Adding attributes on IBM i5/OS
  - Mapping attributes on IBM i5/OS

Perform the following steps to remove an attribute from your database:

**Important:** Do not remove attributes that have already been populated with user values because this can cause database inconsistencies.

**Cluster note:** In a clustered environment, perform the following steps on the deployment manager and then resynch the nodes.

1. Perform the following steps to remove an attribute stored in a property extension database; this is an attribute that was added using the `wp-add-la-property` task.
   A. Open the tool you use to edit your database.
   B. Verify that your attribute name is available in the LAPROP table.
   C. Delete the required attributes from the LAPROP table.
   D. Open the `wimxmlextension.xml` file, located in the `wp_profile_root/config/cells/cellname/wim/model` directory.

   E. Locate and delete the `propertySchema` definition for the attributes that you deleted from the LAPROP table; for example:

   ```xml
   <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String"
   multiValued="true" propertyName="attribute_name">
   <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
   </wim:propertySchema>
   
   </wim:propertySchema>
   
   F. Save your changes to the `wimxmlextension.xml` file.
   G. Open the `wimconfig.xml` file, located in the `wp_profile_root/config/cells/cellname/wim/config` directory.

   H. Locate and delete the `propertiesNotSupported` definitions for the attributes that you deleted from the LAPROP table; for example:

   ```xml
   <config:propertiesNotSupported name="attribute_name"/>
   
   </config:propertiesNotSupported>
   
   I. Save your changes to the `wimconfig.xml` file.
   J. Stop and restart the server1 and WebSphere_Portal servers from the `wp_profile_root/bin` directory.

2. Perform the following steps to remove an attribute that is not stored in a property extension database:
   A. Open the `wimxmlextension.xml` file.
   B. Locate and delete the `propertySchema` definition for the attributes you previously added:

   ```xml
   <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String"
   multiValued="true" propertyName="attribute_name">
   <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
   </wim:propertySchema>
   
   </wim:propertySchema>
   
   C. Save your changes to the `wimxmlextension.xml` file.
   D. Open the `wimconfig.xml` file.

   E. Locate and delete the stanza that corresponds to the custom attribute you deleted from the `wimextension.xml` file; for example:

   ```xml
   <config:attributes name="attribute_name" propertyName="property_name">
   ```
F. Save your changes to the wimconfig.xml file.
G. Stop and restart the server1 and WebSphere_Portal servers.

Parent topic: Adapting the attribute configuration
Previous topic: Mapping attributes on IBM i5/OS
Stand-alone server: Configuring WebSphere Portal to use dynamic groups on i5/OS

By default, WebSphere Portal is enabled for static groups. However, the Virtual Member Manager (VMM) allows users to be members of either static or dynamic groups. Static groups are those where a persistent binding exists between a group and its members. Dynamic groups are those where a search query is defined to retrieve the members of a group. If you have your LDAP server configured to use dynamic groups, complete the steps in this task for WebSphere Portal to use dynamic group queries when you setup your LDAP server.

Perform the required tasks to configure either a stand-alone or federated LDAP server security.

- Prerequisites
  - Preparing a Tivoli Directory Server for IBM i5/OS
  - Choosing your user registry model on IBM i5/OS
  - Adapting the attribute configuration

The steps in this task use groupOfURLs as the object class for dynamic groups and memberURL as the dynamic membership attribute. The actual values for object classes and dynamic membership attributes can vary depending on your LDAP server. For this reason, you should export an LDIF file to verify the object classes and dynamic membership attributes. Either refer to your LDAP documentation or ask your LDAP administrator for instructions on exporting an LDIF file.

Clustered environments: Perform the following steps on the Deployment Manager then synchronize the nodes.

To configure WebSphere Portal to use dynamic groups, do the following:

1. Choose the appropriate set of steps, depending on your LDAP server environment: Table 1. Steps for enabling dynamic groups

<table>
<thead>
<tr>
<th>LDAP server environment</th>
<th>Steps to perform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone LDAP server or federated LDAP server(s)</td>
<td>Navigate to the following directory: <code>wp_profile_root/Cells/cell_name/wim/config</code>. Locate and open <code>wimconfig.xml</code> with any text editor. Add the following line to the <code>&lt;config:groupConfiguration&gt;</code> tag: <code>&lt;config:dynamicMemberAttributes name=&quot;memberurl&quot; objectClass=&quot;groupofurl&quot;/&gt;</code>. Save and close <code>wimconfig.xml</code>.</td>
</tr>
</tbody>
</table>
2. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

**Federated LDAP server(s)**

Log in to the administration console. Select **Security > Secure administration, applications, and infrastructure**. Under **Available realm definitions**, select **Federated repositories** and click **Configure**. Under **Related items**, click **Manage repositories**. Select the appropriate repository from the list. Under **Additional Properties**, click **Group attribute definition** then click **Dynamic member attributes**. Click **New** and specify values for the **Name** and **Object class** fields as appropriate. For example, **Name**: memberurl, **Object class**: groupofurls. Click **OK** and save the changes to the master configuration.

**Parent topic:** Configuring WebSphere Portal to use a user registry on IBM i5/OS  
**Previous topic:** Adapting the attribute configuration  
**Next topic:** Stand-alone server: Enabling referrals for your LDAP user registry on i5/OS
Stand-alone server: Enabling referrals for your LDAP user registry on i5/OS

Referrals redirect object requests from one LDAP server to another when objects do not exist or cannot be located in a particular directory tree. You should enable referrals if your environment has more than one user registry existing on multiple servers or domains.

- **Prerequisites**
  - Preparing a Tivoli Directory Server for IBM i5/OS
  - Choosing your user registry model on IBM i5/OS
  - Adapting the attribute configuration
  - Stand-alone server: Configuring WebSphere Portal to use dynamic groups on i5/OS

To configure your portal to use LDAP referrals, do the following:

1. Use any text editor to open the `wkplc.properties` file in the following directory: `wp_profile_root/ConfigEngine/properties`.

2. Specify values for the following parameters:
   - `et.ldap.id=ID_of_your_LDAP_server`
   - `et.ldap.host=hostname_of_your_LDAP_server`
   - `et.ldap.referral=follow`

3. Save and close `wkplc.properties`.

4. Run the following task from the `wp_profile_root/ConfigEngine` directory to create an LDAP entity type:
   - UNIX: `./ConfigEngine.sh wp-update-et-ldap -DWasPassword=password`
   - Windows: `ConfigEngine.bat wp-update-et-ldap -DWasPassword=password`
   - IBM® i5/OS: `ConfigEngine.sh wp-update-et-ldap -DWasPassword=password`

5. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

**Parent topic:** Configuring WebSphere Portal to use a user registry on IBM i5/OS

**Previous topic:** Stand-alone server: Configuring WebSphere Portal to use dynamic groups on i5/OS
Tune your servers

Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

- Prerequisites
  - Preparing your IBM i5/OS
  - Installing WebSphere Portal on IBM i5/OS
  - Preparing DB2 for i5/OS for a stand-alone production server
  - Configuring WebSphere Portal to use a user registry on IBM i5/OS

Tuning a WebSphere Portal environment involves tuning and configuring the various systems and components of the environment. The tuning guide provides general concepts and details specifics configuration instructions. Instructions are included for:

- Configuring the application server and the resources defined for that application server
- Determining the cloning strategy for expanding or extending the environment
- Tuning the database(s) and database server
- Tuning the directory server and its database
- Tuning the Web server
- Tuning the operating system and network
- Tuning the WebSphere Portal services

Parent topic: Setting up a stand-alone server on i5/OS
Previous topic: Configuring WebSphere Portal to use a user registry on IBM i5/OS

Related information

- WebSphere Portal and Lotus Web Content Management 6.1.x Performance Tuning Guide
- IBM WebSphere Portal Performance Troubleshooting Guide
Setting up a stand-alone server on Linux

After you have prepared the operating system you are ready to install WebSphere Portal.

- Prerequisites
  - Technotes for installation and configuration issues

Select the operating system on which you are installing WebSphere Portal.

1. Preparing your Linux operating system
   View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

2. Installing WebSphere Portal on Linux
   IBM WebSphere Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

3. Configuring databases
   Before transferring default data to the production database server, you need to prepare the database server. Database preparation includes creating user IDs and databases. For this installation path, stand-alone production server, remote databases servers are used.

4. Preparing a remote Web server when portal is installed on Linux
   Install and configure the Web server plug-in, provided by IBM WebSphere Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

5. Configuring WebSphere Portal to use a user registry on Linux
   Configure user registry security on IBM WebSphere Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

6. Tune your servers
   Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

**Parent topic:** Setting up a stand-alone production server
Preparing your Linux operating system

View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

Perform the following steps to prepare your Linux machine:

1. Perform the steps to prepare your operating system for the IBM WebSphere Application Server installation that comes with WebSphere Portal; see one of the following links for information.
   - WebSphere Application Server Version 6.1: Preparing Linux systems for installation
   - WebSphere Application Server Version 7.0: Preparing Linux systems for installation

2. Set the file descriptor limit to 10240; for example, `ulimit -n 10240`.

3. **Web Content Management only:** Use the `ulimit -f` command to set the maximum size of files that can be created to be at least the size of the largest file you would need to upload to the content server. The command `ulimit -f unlimited` removes any limit on file size.

4. Install and configure X server on Linux (for example X-Windows or GNOME) to use the graphical user interface the installation program provides. **Note:** X server is not required if installing with a response file or in console mode.

Parent topic: Setting up a stand-alone server on Linux
Next topic: Installing WebSphere Portal on Linux
Installing WebSphere Portal on Linux

IBM® WebSphere® Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

Review the Supported hardware and software requirements for your operating system and ensure that all hardware and software matches the minimum product level before installing WebSphere Portal. Review Installation methods, options, and sources before installing to determine your installation source location and installation method. Note: If you plan to use IBM WebSphere Application Server Version 7.0, you must purchase and install it before installing WebSphere Portal on the existing WebSphere Application Server Version 7.0 instance.

- Prerequisites
  - Preparing your Linux operating system

Perform the following steps to install WebSphere Portal:

1. Type `ping yourserver.yourcompany.com` on a command line to verify that your fully qualified host name is properly configured.
2. Type `ping localhost` on a command line to verify that your network is properly configured.
3. If you are installing with an existing WebSphere Application Server instance, ensure it is installed at the supported level and has the following features installed:
   - Restriction: Installing WebSphere Portal Version 6.1.0.1 or higher on an existing WebSphere Application Server Version 7.0 instance where WebSphere Portal Version 7.0 was installed is unsupported.
   - Important: Besides ensuring that the existing WebSphere Application Server instance is at the supported level, you will also need to ensure that Apache Derby is installed at the supported level before installing WebSphere Portal. See WebSphere Portal detailed system requirements for supported levels.

- Application Server
- Administration
  - Scripted Administration
  - Administrative Console
- Ant and Deployment Tools
  - Deploy Tool
  - Ant Utilities
4. Optional: Perform the following steps if you want to install WebSphere Portal using a non-root user:
   - Restriction: Although it is possible to install WebSphere Application Server as a non-root user, there are limitations to this option that can affect WebSphere Portal functionality, which is why you should install WebSphere Application Server as a root user.
   A. Install the current supported version of WebSphere Application Server as a root user.
   B. Open a command prompt and perform the following steps to create a non-root user and to change ownership:

      1. Use the appropriate system commands to create a new group, to create a new user, and to add the new user to the new group.
2. Run the following tasks to change the rights of the non-root user:

   chmod -R g+rw /opt/IBM
   chgrp -R group_name /opt/IBM
   chmod -R g+wr /tmp
   chgrp -R group_name /tmp
   chmod -R g+wr /var/tmp
   chgrp -R group_name /var/tmp

C. Launch the installation program per the next step. During the installation, select the existing WebSphere Application Server using the non-root user and set the installation location to a unique location under the path where non-root permissions were granted.

5. Choose one of the following installation commands based on the installation method you want to use to install WebSphere Portal; see Installation Methods for more information:

   **Advance installation parameters:** To customize your installation, you can add various parameters to your installation commands; for example, you can specify your port information. See Advanced installation parameters for information.

   **Restriction:** When defining your installation path, the **ONLY** supported characters are ASCII letters [A-Z and a-z], numbers [0-9], slashes [/ or \
, depending on your operating system], and the dash [-].

<table>
<thead>
<tr>
<th>Table 1. Installation task options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Installation type</strong></td>
</tr>
<tr>
<td>Graphical user interface</td>
</tr>
<tr>
<td>Console mode</td>
</tr>
<tr>
<td>Silent install</td>
</tr>
</tbody>
</table>

**Note:** If the installation program does not detect a WebSphere Application Server instance that you know exists, exit the installation program and pass the WebSphere Application Server instance location using the command line; for example, ./install.sh -W was.undetectedWas="/my/WAS/location".

**Note:** If the GUI or console mode installation program fails to detect ports for either WebSphere Application Server or WebSphere Portal, a warning message displays and the installer offers another chance to enter the values. If the silent installation fails to detect ports for either WebSphere Application Server or WebSphere Portal, the installer will exit.

6. Verify your installation was successful; access WebSphere Portal using the http://yourserver:yourport/wps/portal format. Run the ./ConfigEngine.sh list-server-ports -DWasPassword=password task from the wp_profile_root/ConfigEngine directory to generate the wp_profile_root/ConfigEngine/log/wp_PortMatrix.txt file that lists WebSphere Portal ports for your installation.

7. Optional: After the WebSphere Portal installation completes successfully, run the ./ConfigEngine.sh configure-express -DPortalAdminPwd=password -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, if you want to enable the sample IBM Lotus Web Content Management content, such as internet and intranet sample sites.

   **Restriction:** Run the configure-express task before configuring your database, user registry, context root, security, etc. If you ran any tasks other than the install task, do not run this task.

**Note:** See http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_ovw.html for...
tutorials on how to use the sample content.

The sample content includes:

- Creates a group called contentAuthors; members of this group are given privileges to create content in the sample Internet and intranet sites. Navigate to the **Administration** area and then click **Access > Users and Groups**.

- Creates two new Web Content Management Libraries: "Internet Web Content 6.1.0" and "Intranet Web Content 6.1.0". Navigate to the **Administration** area and then click **Portal Content > Web Content Libraries**.

- Adds a portlet filter and applies the filter to various portlets in the sample Internet and intranet sites. You can see the definition of the filter in the WebSphere Application Server Administration console and examining the custom resources under the **Environment** area.

- Creates two new theme policies: InternetStyle and IntranetStyle. These styles are applied to sample Internet and intranet sites. Navigate to **Theme Customizer** and then select the style.

- Creates several portlet clones of the Web Content Management rendering portlet. These portlet clones are used on sample Internet and intranet sites.

- Creates two virtual portals with context roots of wps/portal/intranet and wps/portal/internet. These are the sample Internet and intranet sites. Go to http://yourserver:yourport/wps/portal/internet and http://yourserver:yourport/wps/portal/intranet to access them.

- Creates several sample credential slots, including "Default slot for E-mail", "Default slot for Feeds", "Default slot for Miscellaneous", "Default slot for Web Clipping", and "Default slot for Web Content Management". Navigate to the **Administration** area and then click **Access > Credential Vault > Manage System Vault Slots**.

- Modifies the portlet palette to include some new portlets. Click the **Portlets** link to see the portlet palette.

8. **Optional:** If you ran the **configure-express** task, the owner of the items in the Web content libraries containing the Internet and Intranet Site Template content will be listed as uid=xyzadmin,o=defaultWIMFileBasedRealm. To update the owner information for these items to correspond to your portal administrator ID, complete the following steps:

   A. Edit the **wp_profile_root/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties** file.

   B. Add the following line to the file:
      ```
      uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
      
      Replace **portal_admin_DN** with the distinguished name of the portal administrator.
      ```

   C. Save your changes and close the file.

   D. Run the **./ConfigEngine.sh action-express-memberfixer -DPortalAdminPwd=password -DWasPassword=password** task, located in the **wp_profile_root/ConfigEngine** directory.

9. **Optional:** Perform the following steps if you need to change the ports for WebSphere Application Server or WebSphere Portal:

   **Note:** The starting port parameter is required for a successful completion of the **modify-ports-by-startport** task.

   Once you specify a start port, this port becomes the base for assigning port values. The code increments this value as each port is assigned, which means that the WebSphere Portal ports will be assigned incrementally starting with the port defined with the **modify-ports-by-startport** task.

   A. Stop the server1 and WebSphere_Portal servers.

   B. Run one of the following commands for each server you need to change:

<table>
<thead>
<tr>
<th>Method</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting port number</td>
<td>./ConfigEngine.sh modify-ports-by-startport -DWasPassword=password -DModifyPortsServer=servername -DStartPort=starting port number</td>
</tr>
</tbody>
</table>
C. Restart the server1 and WebSphere_Portal servers.

### Port file Note:
Sample port files are available on the Setup disc.

```
./ConfigEngine.sh modify-ports-by-portsfile -
DWasPassword=password -
DModifyPortsServer=servername -
DPortsFile=full path to ports file
```

The following is an example of the information within a port file although the port values will be different based on your environment:

```
BOOTSTRAP_ADDRESS=10031
SOAP_CONNECTOR_ADDRESS=10033
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=10032
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=10025
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=10036
WC_adminhost=10027
WC_defaulthost=33344
DCS_UNICAST_ADDRESS=10029
WC_adminhost_secure=10039
WC_defaulthost_secure=10035
SIB_ENDPOINT_ADDRESS=10026
SIB_ENDPOINT_SECURE_ADDRESS=10037
SIB_MQ_ENDPOINT_ADDRESS=10030
SIB_MQ_ENDPOINT_SECURE_ADDRESS=10028
ORB_LISTENER_ADDRESS=10034
```

### Related concepts
- Installation methods
- IBM Support Assistant Lite for WebSphere Portal

### Related tasks
- Creating and maintaining multiple profiles on Linux
- Using the member fixer tool with IBM Lotus Web Content Management

### Related reference
- Advanced installation parameters
Configuring databases

Before transferring default data to the production database server, you need to prepare the database server. Database preparation includes creating user IDs and databases. For this installation path, stand-alone production server, remote database servers are used.

- Prerequisites
  - Preparing your Linux operating system
  - Installing WebSphere Portal on Linux

Select the database server that are using.

- Preparing DB2 on AIX for a stand-alone production server
  You can install DB2 on the same server as WebSphere Portal, also referred to as local, or on a different server, referred to as remote. If you plan to use a local DB2 database, scripts are provided to simplify the database creation and transfer. More information about the local database is provided in the single server installation instructions. For a remote database you must complete the transfer manually.

- Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server
  You can install DB2 on the same server as WebSphere Portal, also referred to as local, or on a different server, referred to as remote. If you plan to use a local DB2 database, scripts are provided to simplify the database creation and transfer. More information about the local database is provided in the single server installation instructions. For a remote database you must complete the transfer manually.

- Preparing DB2 for z/OS for a stand-alone production server
  Setting up the DB2 for z/OS database includes creating user IDs, databases, and table spaces on a remote server.

- Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server
  Setting up the Oracle database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

- Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server
  Setting up the Oracle RAC database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

- Setting up a remote SQL Server database on Windows for a stand-alone production server
  To setup a remote Microsoft SQL Server Enterprise Edition database, create user IDs and databases on a remote server. Tasks are provided to assist with creating the user IDs and the databases. Before you can use the tasks, you must modify properties files.

- Verifying databases
  After you configure IBM WebSphere Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.

Parent topic: Setting up a stand-alone server on Linux

Previous topic: Installing WebSphere Portal on Linux

Next topic: Preparing a remote Web server when portal is installed on Linux
Preparing DB2 on AIX for a stand-alone production server

You can install DB2 on the same server as WebSphere Portal, also referred to as local, or on a different server, referred to as remote. If you plan to use a local DB2 database, scripts are provided to simplify the database creation and transfer. More information about the local database is provide in the single server installation instructions. For a remote database you must complete the transfer manually.

1. **Installing DB2**
   View information on installing DB2 for use with WebSphere Portal.

2. **Create users**
   View the steps to set up users for DB2 to work with WebSphere Portal.

3. **Creating remote databases**
   A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

4. **Optional: Assigning custom table spaces**
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

5. **Modifying database properties**
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

6. **Configuring JCR collation support**
   View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

7. **Configure WebSphere Portal to use DB2**
   View information on how to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

8. **Configuring DB2 for large file handling in Web Content Management**
   If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the fullyMaterializeLobData property in the WebSphere Application Server administrative console.

9. **Changing driver types**
   WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

**Parent topic:** Configuring databases
Installing DB2

View information on installing DB2 for use with WebSphere Portal.

Before you begin:
- Make sure DB2 is set up with updated kernel parameters according to the recommendations in the DB2 Quick Beginnings guide at DB2 Technical Support.
- **(For Type 2 drivers only):** DB2 Connect must be installed on the same system as WebSphere Portal, and have the same name as the server profile name.
- When you install DB2 using the DB2 installation program, it automatically creates a DB2 administrative user with the correct operating system rights.

1. If you are using the JDBC driver in type 2 mode, configure your DB2 Connect client with the following commands. If you are using a remote database, complete this step separately from the WebSphere Portal installation.
   - `db2 update dbm cfg using tp_mon_name WAS`
   - `db2 update dbm cfg using spm_name hostname`, where `hostname` is the host name of WebSphere Portal.

   Because the default for `spm_name` is the hostname itself, specifying the `hostname` parameter is optional. If your hostname is more than eight characters, use empty double quotes (" "). For example, `db2 update dbm cfg using spm_name " "`.

2. Before installing DB2, log in with a user ID that has administrative authority.

3. To install DB2, follow the instructions that are provided with the DB2 documentation to install DB2 and the required fix pack.

Parent topic: Preparing DB2 on AIX for a stand-alone production server

Next topic: Create users
Create users

View the steps to set up users for DB2 to work with WebSphere Portal.

Before you begin: You should have completed Installing DB2.

- **Prerequisites**
  - Installing DB2

For all databases, use one user with administrative rights on the operating system and the DB2 installation. This user can be the database administrative user that is created automatically by the DB2 installation program. This user is the database configuration user that will be used for configuration tasks: creating database tables and performing database transfer. If you choose to have WebSphere Portal also create the databases, the database user should be given **SYSADM** rights. You only need to manually create an administrator ID when you do not want to use an existing DB2 administrator ID.

A common user name is `db2inst1`, but you can assign any user name as long as it has administrative access and follows the limitations listed here. Do not change the user name after creating it.

The limitations on user names are:

- User names can contain one to eight characters.
- Group and instance names can contain one to eight characters.
- Names cannot be any of the following: `users`
  - `admins`
  - `guests`
  - `public`
  - `local`
- Names cannot begin with: `IBM`
  - `SQL`
  - `SYS`
- Names cannot include accented characters.
- Create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Following are the basic steps to create new users and grant privileges. For more information, refer to the DB2 product documentation.

1. From the DB2 Control Center, expand the object tree until you find the DB Users folder.
2. Right-click the DB Users folder and select **Create/Add**. The Create DB User notebook opens.
3. In the User name field, type a user ID.
4. Assign an authority for the user.
5. Click **OK**.
Creating remote databases

A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

Before creating these databases, note the following information:

- Use IBM® DB2 Universal Database™ Enterprise Server Edition JDBC driver, JCC, type 4.
- If the DB2 JDBC type 4 driver is used, only the instructions on the remote server are required. You do not need to install the DB2 client software, and you do not need to complete the steps related to the DB2 client.
- The client software, DB2 Connect, must be correctly configured to connect to the remote DB2 server instance, for example, `db2inst1`.
- These instructions assume that a remote DB2 server and DB2 Connect are already installed and running. For information on installing DB2 Connect, see the DB2 product documentation.

**Prerequisites**
- Installing DB2
- Create users

To create a database, you must be a DB2 System Administrator with sufficient database privileges (SYSADM or at a minimum SYSCTRL).

1. Log in to the DB2 server system as a user with sufficient database privileges.
2. Initialize a DB2 command environment by opening a command prompt and typing `su - db2inst1`, where `db2inst1` is the database user. The prompt changes to your operating system shell prompt, for example: `$`. In this mode, you must type `db2` at the beginning of each command and use double quotation marks (``) around the entire command. The following example shows how to use the double quotes; it is not an actual command that you run:

   ```
   db2 "CREATE REGULAR TABLESPACE SMS PAGESIZE 4 K MANAGED BY SYSTEM USING ('sms') BUFFERPOOL SMSPOOL"
   ```

   **Note:** If you are using the Command Line Processor (CLP), refer to your DB2 documentation for details. The command prompt is `db2=>`. In this mode, commands can be entered without the `db2` prefix or the double quotation marks. However, the following steps assume you are not using the CLP and are entering commands from your operating system shell prompt, for example: `$`.

3. Run the following commands on the DB2 server system to configure the DB2 database instance:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Description</th>
</tr>
</thead>
</table>
   | DB2         | db2set DB2_RA_TO_RS=YES  
   |             | DB2_EVALUNCOMMITTED=YES  
   |             | db2set DB2_INLIST_TO_NJLN=YES  
   |             | db2 "UPDATE DBM CFG USING query_heap_sz 12768"  
   |             | db2 "UPDATE DBM CFG USING maxagents 500"  
   |             | db2 "UPDATE DBM CFG USING sheaphres 0"  

4. **(Important)** Failure to complete this step will result in an unsuccessful database transfer. Run the following commands on the DB2 server system to create the necessary databases:

   **Notes:**
   - Replace `dbname` with the actual name of the database. Run the commands and each time replace `dbname` with the actual values for release, community, customization, Java Content Repository, Feedback, and Likeminds. You will need to run the commands once for each database for a total of six times.
   - **Remember:** DB2 database names cannot
exceed eight characters. Therefore, consider using these database names: release, commun, custom, jcrdb, fdbkdb, and lmdb.

```
db2 "CREATE DB dbname using codeset UTF-8 territory us PAGESIZE 8192"
db2 "UPDATE DB CFG FOR dbname USING applheapsz 4096"
db2 "UPDATE DB CFG FOR dbname USING app_c1_heap_sz 1024"
db2 "UPDATE DB CFG FOR dbname USING stmheap 32768"
db2 "UPDATE DB CFG FOR dbname USING dbheap 2400"
db2 "UPDATE DB CFG FOR dbname USING locklist 1000"
db2 "UPDATE DB CFG FOR dbname USING logfilsiz 4000"
db2 "UPDATE DB CFG FOR dbname USING logprimary 12"
db2 "UPDATE DB CFG FOR dbname USING logsecond 20"
db2 "UPDATE DB CFG FOR dbname USING logbufsz 32"
db2 "UPDATE DB CFG FOR dbname USING avg_appls 5"
db2 "UPDATE DB CFG FOR dbname using AUTO_MAINT off"
```

**5. Complete the following:**

A. On the DB2 server system, run the following commands. This step is only required for the IBM Java Content Repository database (`jcrdb`).

- `jcrdb` is the name of the database used to store user data and objects
- `-jcr` is the jcr user for `jcrdb`
- `dbpassword` is the password for the jcr user for the `jcrdb`

```
db2 "CONNECT TO jcrdb USER jcr USING dbpassword"
db2 "CREATE BUFFERPOOL ICMLSFREQBP4 SIZE 1000 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL ICMLSVOLATILEBP4 SIZE 16000 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL ICMLSMAINBP32 SIZE 16000 PAGESIZE 32 K"
db2 "CREATE BUFFERPOOL ICMLSFREQBP4 SIZE 1000 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL ICMLSVOLATILEBP4 SIZE 16000 PAGESIZE 4 K"
```

**Note:** This value can be replaced with any ID that has administrative authority.

B. For JDBC Type 2 connections only: On DB2 Connect, use a text editor to open the `/etc/services` file. If it does not specify the DB2 connection service port, add the following text to specify the port for the remote DB2 instance:

```
db2c_{db2inst1}/tcp # DB2 connection service port
```

where `db2inst1` is the name of the DB2 instance ID on the system, and `port1` with the actual port number that is
assigned to the DB2 connection service in your DB2 server installation. The connection service port on the DB2 Client system, WebSphere Portal server, must match the connection service port on the DB2 server. The ports should match by number but not necessarily by name.

C. Set up the correct service name by entering the following command on the DB2 server system: `db2 "UPDATE DBM CFG USING svcename svce_name"` where `svce_name` is the connection service port name that is specified above.

D. For JDBC Type 2 connections only: On DB2 Connect, set DB2COMM to TCP/IP by using the `db2set` command `db2set DB2COMM=tcpip`.

E. For JDBC Type 2 connections only: Catalog the TCP/IP node with the IP address of the remote database server on DB2 Connect. `db2 "catalog tcpip node remote_db_node_alias remote database_server_node server connection_service_port"` where:
   - `remote_db_node_alias` is the alias name of the database server that you are defining for the WebSphere Application Server node name. The alias name can contain one to eight characters.
   - `database_server_node` is the fully qualified host name of your database server system.
   - `connection_service_port` is the name of the DB2 connection service port that is configured in the `/etc/services` file on the database server system.

F. For JDBC Type 2 connections only, catalog the WebSphere Portal databases on DB2 Connect, where:
   - `remote_db_name_domain`, is the cataloged name of the databases on the server system for each domain.
   - `domain_alias_name`, is the database alias names that you are defining.
   - `remote_db_node_alias` is the name that was used previously when you cataloged the TCP/IP node in the previous step.

   The alias for each database must be different from the actual database name and can only contain up to eight characters:
   `db2 "catalog db remote_db_name_release as release_alias_name at node remote_db_node_alias"
   db2 "catalog db remote_db_name_community as comm_alias_name at node remote_db_node_alias"
   db2 "catalog db remote_db_name_customization as cust_alias_name at node remote_db_node_alias"
   db2 "catalog db remote_db_name_fdbkdb as fdbkdb_alias_name at node remote_db_node_alias"
   db2 "catalog db remote_db_name_lmdb as lmdb_alias_name at node remote_db_node_alias"
   db2 "catalog db remote_db_name_jcrdb as jcrdb_alias_name at node remote_db_node_alias"

G. For JDBC Type 2 connections only: Log out of DB2 Connect by entering `db2 "terminate"`.

H. For JDBC Type 2 connections only, on DB2 Connect, test your remote connection by issuing the following command in the DB2 command window: `db2 "connect to alias_name user username using password"`, where `alias_name` is the alias name that you defined above, `username` is the database user, and `password` is the password assigned to the database user.

**Parent topic:** Preparing DB2 on AIX for a stand-alone production server  
**Previous topic:** Create users  
**Next topic:** Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root` /PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- The page size of table spaces used by WebSphere Portal must be 8192 bytes.
- For details on creating table spaces refer to the documentation for the database.

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`

   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds

3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword `IN` and a space. For example: `community.COMP_INST.tablespace=IN COMM8KSPACE`

   Repeat this step for each domain that you are transferring.

4. Save and close `dbdomain.space_mapping.properties`.

5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
   - UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
- `i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

**Parent topic:** Preparing DB2 on AIX for a stand-alone production server  
**Previous topic:** Creating remote databases  
**Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:
  - `release.DbName=release`
  - `jcr.DbName=jcrdb`
  - `feedback.DbName=fdbkdb`
  - `likeminds.DbName=lmdb`
  - `community.DbName=commdb`
  - `customization.DbName=customdb`

  - If you are using a remote database, enter the values for the remote server.
  - Use a forward slash (/) instead of a backslash (\).
  - There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
  - The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.

- Depending on which database domain has to be configured, replace `dbdomain` with:
  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`

- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:

   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.

   A. For dbdomain.DbType, type db2.

   B. For dbdomain.DbName, type the name of the WebSphere Portal domain database. **Notes:**

      - This value is also the database element in the dbdomain.DbUrl property.

      - This value is the TCP-IP alias for the database.

   C. For dbdomain.DbSchema, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.

   D. For dbdomain.DataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:

      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback

   E. For dbdomain.DbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:** The database element of this value should match the value of DbName.

   F. For dbdomain.DbUser, type the user ID for the database administrator.

   G. For dbdomain.DbPassword, type the password for the database administrator.

   H. Optional: For dbdomain.DbRuntimeUser, type the user ID of the database user that should be used by WebSphere Portal to connect to the database at runtime.

   I. If dbdomain.DbRuntimeUser is specified, you must set dbdomain.DbRuntimePassword to be the password of the runtime database user.

   J. For dbdomain.DBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.
K. For **dbdomain.DBA.DbPassword**, type the database administrator password for privileged access operations during creation of the database.

L. For **dbdomain.XDbName**, type the database loop back alias that needs to be set if you plan to use the `create-database` task. **Note**: Required only for local databases using a JDBC Type 2 driver.

M. For **dbdomain.DbNode**, type the value for the node database. Set this value if you want to call `create-database`. **Note**: Required only for local databases.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important**: The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For **source.domain.DbType**, type of the database you are currently configured to use. The value for **source.domain(DbType)** is Derby by default.

B. For **source.domain.DbName**, type the name of the database domain you are currently using.

C. For **source.domain.DbSchema**, type current schema identifier for objects within the database for this domain.

D. For **source.domain.DataSourceName**, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For **source.domain.DbUrl**, type the url currently used to access your database.

F. For **source.domain.DbUser**, type the name of the user accessing this database.

G. For **source.domain.DbPassword**, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file **wkplc_dbtype.properties**.

A. For **db2.DbDriver**, type the name of the JDBC driver class.

B. For **db2.DbLibrary**, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

C. For **db2.JdbcProviderName**, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file **wkplc.properties**.

A. For **WasPassword**, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic**: Preparing DB2 on AIX for a stand-alone production server  
**Previous topic**: Assigning custom table spaces  
**Next topic**: Configuring JCR collation support
Configuring JCR collation support

View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties

1. Stop the WebSphere Portal server:
   ```
   ./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password
   ```

2. If using a remote DB2 server, copy the following files from the WebSphere Portal server to a temporary directory on the remote DB2 server:
   ```
   PortalServer_root/jcr/prereq.jcr/config/collation.jar
   PortalServer_root/jcr/prereq.jcr/config/registerCollationUDFTemplate.sql
   ```

3. Set up collation on the database where the JCR domain is located.
   A. Change to the directory `db2_instance_owner_home/sqllib/function`.
   B. Execute the command:
      ```
      Remote DB2:
      db2_instance_owner_home/sqllib/java/jdk/bin/jar -xvf temporary\location/collation.jar icm/CollationUDF.class
      Local DB2:
      db2_instance_owner_home/sqllib/java/jdk/bin/jar -xvf PortalServer_root\jcr/prereq.jcr/config/collation.jar icm/CollationUDF.class
      ```
   C. If using a local DB2, change to the directory `wp_profile_root/PortalServer/jcr/config`. If using a remote DB2, change to the temporary directory where you copied the files in a previous step.
   D. Open the file `registerCollationUDFTemplate.sql` and change all `SCHEMA` references to the JCR schema; for example, `JCR`. The value set for `SCHEMA` should match the value set for the `jcr.DbSchema` property. You specify `jcr.DbSchema` in the configuration file `wkplc_comp.properties` when you modify database properties.

4. Connect to the JCR database by running `db2 connect to <jcrdb> user <userid> using <password>`. You must connect to the JCR database using the userid and password of the WebSphere Portal JCR data source.

5. Edit the `icm.properties` file, located in `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm` directory. Add the following section to the end of the file:

   ```
   # Enable/Disable collation support for all DB2 platforms
   ```
# Disabled by default
jcr.query.collation.db2.enabled = true

# Database specific collation mappings
# These mappings apply map a Java locale name into a collation name
# supported by the underlying database.
# Example mappings for DB2 platform

# English
jcr.query.collation.en = en

# Swedish
jcr.query.collation.sv = sv

jcr.query.collation.zh = zh
jcr.query.collation.de = de
jcr.query.collation.da = da
jcr.query.collation.hu = hu
jcr.query.collation.jp = jp

6. Enter the following command to start the WebSphere Portal server:
   `./startServer.sh WebSphere_Portal`

**Parent topic:** Preparing DB2 on AIX for a stand-alone production server  
**Previous topic:** Modifying database properties  
**Next topic:** Configure WebSphere Portal to use DB2
Configure WebSphere Portal to use DB2

View information on how to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

**Prerequisites**
- Installing DB2
- Create users
- Creating remote databases
- Modifying database properties

**Tips:**
- To run these tasks as a non-root user, you must first run the task chown -R non-root_user WebSphereDir.
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

**Steps for transferring data to another supported database**

1. If you are running a type 2 connection, before transferring data edit the db2cli.ini file. Failure to follow these steps will cause the database transfer to hang at the task action-process-constraints.
   A. Locate the file /home/db2inst1/sqlib/cfg/db2cli.ini.
   B. Add the following to the end of the file. Leave an empty line after ReturnAliases=0.

   ```
   [COMMON]
   DYNAMIC=1
   ReturnAliases=0
   ```

2. Perform this step only if you are installing multiple instances of WebSphere Portal. Change the maximum number of databases MAX_NETBIOS_CONNECTIONS to increase the default configured number of databases. For example, enter the following command at the database prompt: set client MAX_NETBIOS_CONNECTIONS 254 A message indicates success if the number was increased.

3. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.

4. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2.

   ```
   /home/db2inst1/sqlib/db2profile
   where db2inst1 represents your database instance
   ```

   **Note:** You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.

5. Enter the following commands to validate configuration properties.

   ```
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ```
6. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.

7. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td><code>.stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td><code>.stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

8. Transfer the database: **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory `wp_profile_root/ConfigEngine`.

   B. Enter the following command: `.ConfigEngine.sh database-transfer -DWasPassword=password`

   **Note:**
   - To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command: `.ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password`
   - If you have been storing data in IBM® Derby for a long time, database transfer could fail with OutOfMemory exceptions. If database transfer fails, add the following property to the command in this step: `.ConfigEngine.sh database-transfer -DDbtJavaMaxMemory=1536M -DWasPassword=password`

   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

9. After transferring the database tables, perform a `reorg` check to improve performance. Perform this step for each database alias in the property file.

   A. Connect to a database with the following command:

   ```
   db2 connect to database_alias user db2admin_userid using password
   ```

   **Note:** Additional options might be required if additional security has been installed. Refer to DB2 Universal Database commands by example for links to the command reference.

   B. After it is connected, run the following commands from the DB2 prompt:

   ```
   db2 reorgchk update statistics on table all > xyz.out
   ```

   C. Look in the reorg column for entries marked with a * (star or asterisk) in the file `xyz.out`. For each line with a *, note the `tablename` and run the following command for each `tablename`:

   ```
   db2 reorg table tablename db2 terminate db2rbind
database_name -l db2rbind.out -u db2_admin -p password
   ```

   D. The output file `db2rbind.out` is only created when there is an error for the `db2rbind` command.

10. Change to the directory `wp_profile_root/bin`.

11. Enter the following command to start the WebSphere Portal server: `.startServer.sh WebSphere_Portal`

Parent topic: Preparing DB2 on AIX for a stand-alone production server

Previous topic: Configuring JCR collation support

Next topic: Configuring DB2 for large file handling in Web Content Management
Configuring DB2 for large file handling in Web Content Management

If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties
  - Configure WebSphere Portal to use DB2

  **Note:** You only need to perform these steps if you are using Web Content Management.

1. Log into the WebSphere Application Server administrative console.
2. Click Resources > JDBC > Data sources.
3. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
4. Select the name of the data source that is defined in `wkplc_comp.properties` for the JCR database domain. The default data source is `wpdbDS`.
5. Click Custom properties.
6. Ensure that the `fullyMaterializeLobData` property is set to false.

**Parent topic:** Preparing DB2 on AIX for a stand-alone production server  
**Previous topic:** Configure WebSphere Portal to use DB2  
**Next topic:** Changing driver types
Changing driver types

WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Before you begin, ensure that the following conditions are met:
- The WebSphere Portal database has been successfully transferred to DB2 using the `database-transfer` configuration task.
- The files `wkplc_comp.properties` and `wkplc_dbtype.properties` have been modified to set the correct values for the DB2 drivers that you are switching to:

  In the file `wkplc_comp.properties` set each `<Domain>.DbUrl` property using the following formats:
  
  # db2 (type 2):        { jdbc:db2:wpsdb }
  # db2 (type 4):        { jdbc:db2://<YourDatabaseServer>:50000/wpsdb:returnAlias=0; }

  In the file `wkplc_dbtype.properties` set the `db2.DbLibrary` property using the following format:
  
  # For DB2 Type 2 driver use <SQLLIB>/java/db2java.zip
  # For DB2 Type 4 driver use <SQLLIB>/java/db2jcc.jar:<SQLLIB>/java/db2jcc_license_cu.jar

  In the file `wkplc_dbtype.properties` set the `db2.DbDriver` property using the following format:
  
  # For DB2 Type 2 driver use COM.ibm.db2.jdbc.app.DB2Driver
  # For DB2 Type 4 driver use com.ibm.db2.jcc.DB2Driver

**Note:** If WebSphere Portal is installed on the same machine as the DB2 server and you switch from a JDBC Type 4 connection to a JDBC Type 2 connection, verify that you have created the alias names for the DB2 databases as described in Creating remote databases and that the alias names are specified for the databases in the file `wkplc_comp.properties`. When switching from a JDBC Type 2 connection to a JDBC Type 4 connection, remove the database alias names and refer to the databases directly. This is required because of a limitation in the DB2 Universal JDBC driver.

**Prerequisites**

- Installing DB2
- Create users
- Creating remote databases
- Modifying database properties
- Configure WebSphere Portal to use DB2

1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.
2. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2.

   ```
   /home/db2inst1/sqlib/db2profile
   ```

   where `db2inst1` represents your database instance **Note:** You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.
3. Enter the following commands to validate configuration properties.

   ```
   . /ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   . /ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,
   ```
4. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td><code>./stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td><code>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

6. Change to the directory `wp_profile_root/ConfigEngine`.

7. To change from one supported driver to the other, run the following task to connect the database, including only the domains that require the switch:

   ```
   ```

8. Change to the directory `wp_profile_root/bin`.

9. Enter the following command to start the WebSphere Portal server:

   ```
   ./startServer.sh WebSphere_Portal
   ```

---

**Parent topic:** Preparing DB2 on AIX for a stand-alone production server  
**Previous topic:** Configuring DB2 for large file handling in Web Content Management
Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

You can install DB2 on the same server as WebSphere Portal, also referred to as local, or on a different server, referred to as remote. If you plan to use a local DB2 database, scripts are provided to simplify the database creation and transfer. More information about the local database is provided in the single server installation instructions. For a remote database you must complete the transfer manually.

1. Installing DB2
   View information on installing DB2 for use with WebSphere Portal.

2. Create users
   View the steps to set up users for DB2 to work with WebSphere Portal.

3. Creating remote databases
   A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

4. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

5. Modifying database properties
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

6. Configuring JCR collation support
   View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

7. Configure WebSphere Portal to use DB2
   View the steps to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

8. Configuring DB2 for large file handling in Web Content Management
   If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the fullyMaterializeLobData property in the WebSphere Application Server administrative console.

9. Changing driver types
   WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Parent topic: Configuring databases
Installing DB2

View information on installing DB2 for use with WebSphere Portal.

Before you begin:
- Make sure DB2 is set up with updated kernel parameters according to the recommendations in the DB2 Quick Beginnings guide at DB2 Technical Support.
- **(For Type 2 drivers only):** DB2 Connect must be installed on the same system as WebSphere Portal, and have the same name as the server profile name.
- When you install DB2 using the DB2 installation program, it automatically creates a DB2 administrative user with the correct operating system rights.

1. If you are using the JDBC driver in type 2 mode, configure your DB2 Connect client with the following commands. If you are using a remote database, complete this step separately from the WebSphere Portal installation.
   - `db2 update dbm cfg using tp_mon_name WAS`
   - `db2 update dbm cfg using spm_name hostname`, where `hostname` is the host name of WebSphere Portal.
     Because the default for `spm_name` is the hostname itself, specifying the `hostname` parameter is optional. If your hostname is more than eight characters, use empty double quotes (" "). For example, `db2 update dbm cfg using spm_name " "`.

2. Before installing DB2, log in with a user ID that has administrative authority. The system user ID and password must match the database user ID and password for the WebSphere Portal databases. If you do not use the default DB2 database user ID, or if you must access a remote database, create the system user ID before installing DB2.
   To set up the environment for the database access:
   A. The initialization script for this user (for example, `user-home/.profile`) must contain a call to the `db2profile` script in the `db-home/sqllib` directory.
   B. After you create the system user ID and password for the DB2 installation, add the user ID to the DB2 administration group (such as `db2admin`) for that system.

   To ensure that the user is set up correctly, login with this user ID and start the DB2 command line processor by executing the command: `db2`. If the command line processor displays, the user ID is set up correctly.

3. To install DB2, follow the instructions that are provided with the DB2 documentation to install DB2 and the required fix pack.
Create users

View the steps to set up users for DB2 to work with WebSphere Portal.

Before you begin: You should have completed Installing DB2.

- Prerequisites
  - Installing DB2

For all databases, use one user with administrative rights on the operating system and the DB2 installation. This user can be the database administrative user that is created automatically by the DB2 installation program. This user is the database configuration user that will be used for configuration tasks: creating database tables and performing database transfer. If you choose to have WebSphere Portal also create the databases, the database user should be given SYSADM rights. You only need to manually create an administrator ID when you do not want to use an existing DB2 administrator ID.

A common user name is `db2inst1`, but you can assign any user name as long as it has administrative access and follows the limitations listed here. Do not change the user name after creating it.

The limitations on user names are:

- User names can contain one to eight characters.
- Group and instance names can contain one to eight characters.
- Names cannot be any of the following: `users` `admins` `guests` `public` `local` `IBM` `SQL` `SYS` `SYSADM`
- Names cannot include accented characters.
- Create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Following are the basic steps to create new users and grant privileges. For more information, refer to the DB2 product documentation.

1. From the DB2 Control Center, expand the object tree until you find the DB Users folder.
2. Right-click the DB Users folder and select Create/Add. The Create DB User notebook opens.
3. In the User name field, type a user ID.
4. Assign an authority for the user.
5. Click OK.

Parent topic: Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server
Previous topic: Installing DB2
Next topic: Creating remote databases
Creating remote databases

A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

Before creating these databases, note the following information:

- Use IBM® DB2 Universal Database™ Enterprise Server Edition JDBC driver, JCC, type 4.

- If the DB2 JDBC type 4 driver is used, only the instructions on the remote server are required. You do not need to install the DB2 client software, and you do not need to complete the steps related to the DB2 client.

- The client software, DB2 Connect, must be correctly configured to connect to the remote DB2 server instance, for example, db2inst1.

- These instructions assume that a remote DB2 server and DB2 Connect are already installed and running. For information on installing DB2 Connect, see the DB2 product documentation.

- **Prerequisites**
  - Installing DB2
  - Create users

To create a database, you must be a DB2 System Administrator with sufficient database privileges (SYSADM or at a minimum SYSCTRL).

1. Log in to the DB2 server system as a user with sufficient database privileges.
2. Initialize a DB2 command environment by opening a command prompt and typing su - db2inst1, where db2inst1 is the database user. The prompt changes to your operating system shell prompt, for example: $. In this mode, you must type db2 at the beginning of each command and use double quotation marks (") around the entire command. The following example shows how to use the double quotes; it is not an actual command that you run:

   db2 "CREATE REGULAR TABLESPACE SMS PAGESIZE 4 K MANAGED BY SYSTEM USING ('sms') BUFFERPOOL SMSPOOL"

   **Note:** If you are using the Command Line Processor (CLP), refer to your DB2 documentation for details. The command prompt is db2=>. In this mode, commands can be entered without the db2 prefix or the double quotation marks.

   However, the following steps assume you are not using the CLP and are entering commands from your operating system shell prompt, for example: $.

3. Run the following commands on the DB2 server system to configure the DB2 database instance:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td>db2set DB2_RA_TO_RS=YES &lt;br&gt; db2set DB2_EVALUNCOMMITTED=YES &lt;br&gt; db2set DB2_INLIST_TO_NLJN=YES &lt;br&gt; db2 &quot;UPDATE DBM CFG USING query_heap_sz 32768&quot; &lt;br&gt; db2 &quot;UPDATE DBM CFG USING maxagents 500&quot; &lt;br&gt; db2 &quot;UPDATE DBM CFG USING sheapthres 0&quot;</td>
</tr>
</tbody>
</table>

4. **(Important)** Failure to complete this step will result in an unsuccessful database transfer. Run the following commands on the DB2 server system to create the necessary databases:

   - Replace dbname with the actual name of the database. Run the commands and each time replace dbname with the actual values for release, community, customization, Java Content Repository, Feedback, and Likeminds. You will need to run the commands once for each database for a total of six times.

   **Remember:** DB2 database names cannot
exceed eight characters. Therefore, consider using these database names: `release, commun, custom, jcrdb, fdbkdb, and lmdb`.

```
db2 "CREATE DB dbname using codeset UTF-8 territory us PAGESIZE 8192"
db2 "UPDATE DB CFG FOR dbname USING applheapsz 4096"
db2 "UPDATE DB CFG FOR dbname USING app_ctl_heap_sz 1024"
db2 "UPDATE DB CFG FOR dbname USING stmtheap 32768"
db2 "UPDATE DB CFG FOR dbname USING dbheap 2400"
db2 "UPDATE DB CFG FOR dbname USING locklist 1000"
db2 "UPDATE DB CFG FOR dbname USING logfilsiz 4000"
db2 "UPDATE DB CFG FOR dbname USING logprimary 12"
db2 "UPDATE DB CFG FOR dbname USING logsecond 20"
db2 "UPDATE DB CFG FOR dbname USING logbufsz 32"
db2 "UPDATE DB CFG FOR dbname USING avg_appls 5"
db2 "UPDATE DB CFG FOR dbname using AUTO_MAINT off"
```

Complete the following:

5. **On the DB2 server system, run the following commands. This step is only required for the IBM Java Content Repository database (`jcrdb`).**

- `jcrdb` is the name of the database used to store user data and objects
- `jcr` is the jcr user for `jcrdb` **Note:** This value can be replaced with any ID that has administrative authority.
- `dbpassword` is the password for the jcr user for the `jcrdb`

```
db2 "CONNECT TO jcrdb USER jcr USING dbpassword"
db2 "CREATE BUFFERPOOL ICMLSFREQBP4 SIZE 1000 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL ICMLSVOLATILEBP4 SIZE 16000 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL ICMLSMAINBP32 SIZE 16000 PAGESIZE 32 K"
db2 "CREATE BUFFERPOOL CMBMAIN4 SIZE 1000 PAGESIZE 4 K"
db2 "CREATE REGULAR TABLESPACE ICMLFQ32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLFQ32') BUFFERPOOL ICMLSMAINBP32"
db2 "CREATE REGULAR TABLESPACE ICMLNF32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLNF32') BUFFERPOOL ICMLSMAINBP32"
db2 "CREATE REGULAR TABLESPACE ICMVFOQ4 PAGESIZE 4 K MANAGED BY SYSTEM USING ('ICMVFOQ4') BUFFERPOOL ICMLSVOLATILEBP4"
db2 "CREATE REGULAR TABLESPACE ICMSFQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('ICMSFQ04') BUFFERPOOL ICMLSVOLATILEBP4"
db2 "CREATE REGULAR TABLESPACE CMBINV04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('CMBINV04') BUFFERPOOL CMBMAIN4"
```

For JDBC Type 2 connections only: On DB2 Connect, use a text editor to open the `/etc/services` file. If it does not specify the DB2 connection service port, add the following text to specify the port for the remote DB2 instance:

```
db2 "DISCONNECT jcrdb"
db2 "TERMINATE"
```

B. **For JDBC Type 2 connections only:** On DB2 Connect, use a text editor to open the `/etc/services` file. If it does not specify the DB2 connection service port, add the following text to specify the port for the remote DB2 instance:

```
db2 "db2inst1/tcp # DB2 connection service port"
```

where `db2inst1` is the name of the DB2 instance ID on the system, and `port` with the actual port number that is
assigned to the DB2 connection service in your DB2 server installation. The connection service port on the DB2
Client system, WebSphere Portal server, must match the connection service port on the DB2 server. The ports
should match by number but not necessarily by name.

C. Set up the correct service name by entering the following command on the DB2 server system:
db2 "UPDATE DBM
CFG USING svcname svce_name" where svce_name is the connection service port name that is specified above.

D. For JDBC Type 2 connections only: On DB2 Connect, set DB2COMM to TCP/IP by using the db2set command
db2set DB2COMM=tcpip.

E. For JDBC Type 2 connections only: Catalog the TCP/IP node with the IP address of the remote database server on
DB2 Connect: db2 "catalog tcpip node remote_db_node_alias remote database_server_node server
connection_service_port" where:
- remote_db_node_alias is the alias name of the database server that you are defining for the WebSphere Application
  Server node name. The alias name can contain one to eight characters.
- database_server_node is the fully qualified host name of your database server system.
- connection_service_port is the name of the DB2 connection service port that is configured in the /etc/services
  file on the database server system.

F. For JDBC Type 2 connections only, catalog the WebSphere Portal databases on DB2 Connect, where:
- remote_db_name_domain, is the cataloged name of the databases on the server system for each domain.
- domain_alias_name, is the database alias names that you are defining.
- remote_db_node_alias is the name that was used previously when you cataloged the TCP/IP node in the previous
  step.
  The alias for each database must be different from the actual database name and can only contain up to eight
  characters.db2 "catalog db remote_db_name_release as release_alias_name at node remote_db_node_alias"
  db2 "catalog db remote_db_name_community as comm_alias_name at node remote_db_node_alias"
  db2 "catalog db remote_db_name_customization as cust_alias_name at node remote_db_node_alias"
  
  db2 "catalog db remote_db_name_fdbkdb as fdbkdb_alias_name at node remote_db_node_alias"
  db2 "catalog db remote_db_name_lmdb as lmdb_alias_name at node remote_db_node_alias"
  db2 "catalog db remote_db_name_jcrdb as jcrdb_alias_name at node remote_db_node_alias"

G. For JDBC Type 2 connections only: Log out of DB2 Connect by entering db2 "terminate".

H. For JDBC Type 2 connections only, on DB2 Connect, test your remote connection by issuing the following command
in the DB2 command window: db2 "connect to alias_name user username using password", where alias_name
is the alias name that you defined above, username is the database user, and password is the password assigned to
the database user.

Parent topic: Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server
Previous topic: Create users
Next topic: Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root` /PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- The page size of table spaces used by WebSphere Portal must be 8192 bytes.
- For details on creating table spaces refer to the documentation for the database.

**Prerequisites**

- Installing DB2
- Create users
- Creating remote databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`

   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - `release`
   - `community`
   - `customization`
   - `jcr`
   - `feedback`
   - `likeminds`

3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword `IN` and a space. For example: `community.COMP_INST.tablespace=IN COMM8KSPACE`

   Repeat this step for each domain that you are transferring.

4. Save and close `dbdomain.space_mapping.properties`.

5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - Windows: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
   - UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

Parent topic: Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server
Previous topic: Creating remote databases
Next topic: Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:
  - release.DbName = release
  - jcr.DbName = jcrdb
  - feedback.DbName = fdbkdb
  - likeminds.DbName = lmdb
  - community.DbName = commdb
  - customization.DbName = custdb
- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
If you use the same values for all four properties across the release, customization, community, and JCR domains, the
database-transfer task fails because of ambiguous database object names. If \textit{DbUser}, \textit{DbUrl}, and \textit{DbPassword} are not
the same across domains, the value for \textit{DataSourceName} must differ from the \textit{DataSourceName} of the other domains. In
other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   \begin{itemize}
   \item \texttt{wp_profile_root}/ConfigEngine/properties/wkplc.properties
   \item \texttt{wp_profile_root}/ConfigEngine/properties/wkplc_comp.properties
   \item \texttt{wp_profile_root}/ConfigEngine/properties/wkplc_dbtype.properties
   \end{itemize}
   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out
   the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each
   instance of each property. In \texttt{wkplc}\_\texttt{comp}\.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file \texttt{wkplc}\_\texttt{comp}\.properties and modify the values to correspond to your
   environment.
   \begin{enumerate}
   \item For \textit{dbdomain}.DbType, type \texttt{db2}.
   \item For \textit{dbdomain}.DbName, type the name of the WebSphere Portal domain database. \textbf{Notes:}
     \begin{itemize}
     \item This value is also the database element in the \textit{dbdomain}.DbUrl property.
     \item This value is the TCP-IP alias for the database.
     \end{itemize}
   \item For \textit{dbdomain}.DbSchema, type the schema name of the database domain. \textbf{Note:} Review your target database
     management system documentation to define a valid schema name. Some database management systems have
     schema name restrictions that you need to understand.
   \item For \textit{dbdomain}.DataSourceName, type the name of the data source that WebSphere Portal uses to communicate
     with its databases. Do not use the following reserved words:
     \begin{itemize}
     \item releaseDS
     \item communityDS
     \item customizationDS
     \item jcrDS
     \item lmdbDS
     \item feedback
     \end{itemize}
   \item For \textit{dbdomain}.DbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value
     must conform to the JDBC URL syntax specified by the database. \textbf{Note:} The database element of this value should
     match the value of DbName.
   \item For \textit{dbdomain}.DbUser, type the user ID for the database administrator.
   \item For \textit{dbdomain}.DbPassword, type the password for the database administrator.
   \item Optional: For \textit{dbdomain}.DbRuntimeUser, type the user ID of the database user that should be used by WebSphere
     Portal to connect to the database at runtime.
   \item If \textit{dbdomain}.DbRuntimeUser is specified, you must set \textit{dbdomain}.DbRuntimePassword to be the password of the
     runtime database user.
   \item For \textit{dbdomain}.DBA.DbUser, type the database administrator user ID for privileged access operations during creation
     of the database.
   \end{enumerate}
For dbdomain.DBA.DbPassword, type the database administrator password for privileged access operations during creation of the database.

For dbdomain.XDbName, type the database loop back alias that needs to be set if you plan to use the create-database task. **Note:** Required only for local databases using a JDBC Type 2 driver.

For dbdomain.DbNode, type the value for the node database. Set this value if you want to call create-database. **Note:** Required only for local databases.

(Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For source.domain.DbType, type of the database you are currently configured to use. The value for source.domain .DbType is Derby by default.

B. For source.domain.DbName, type the name of the database domain you are currently using.

C. For source.domain.DbSchema, type current schema identifier for objects within the database for this domain.

D. For source.domain.DataSourceName, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For source.domain.DbUrl, type the url currently used to access your database.

F. For source.domain.DbUser, type the name of the user accessing this database.

G. For source.domain.DbPassword, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file wkplc_dbtype.properties.

A. For db2.DbDriver, type the name of the JDBC driver class.

B. For db2.DbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

C. For db2.JdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file wkplc.properties.

A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Assigning custom table spaces

**Next topic:** Configuring JCR collation support
Configuring JCR collation support

View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

**Prerequisites**
- Installing DB2
- Create users
- Creating remote databases
- Modifying database properties

1. Stop the WebSphere Portal server:
   ```sh
   ./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password
   ```

2. If using a remote DB2 server, copy the following files from the WebSphere Portal server to a temporary directory on the remote DB2 server:
   ```
   PortalServer_root/jcr/prereq.jcr/config/collation.jar
   PortalServer_root/jcr/prereq.jcr/config/registerCollationUDFTemplate.sql
   ```
   Set up collation on the database where the JCR domain is located.
   ```
   Change to the directory `db2_instance_owner_home/sqllib/function`.
   Execute the command:
   ```
   Remote DB2:
db2_instance_owner_home/sqllib/java/jdk/bin/jar -xvf temporary location/collation.jar icm/CollationUDF.class
   Local DB2:
db2_instance_owner_home/sqllib/java/jdk/bin/jar -xvf PortalServer_root/jcr/prereq.jcr/config/collation.jar icm/CollationUDF.class
   ```

3. Change to the directory `wp_profile_root/PortalServer/jcr/config`. If using a remote DB2, change to the temporary directory where you copied the files in the previous step.

4. Open the file `registerCollationUDFTemplate.sql` and change all `SCHEMA` references to the JCR schema; for example, `JCR`. The value set for `SCHEMA` should match the value set for the `jcr.DbSchema` property. You specify `jcr.DbSchema` in the configuration file `wkplc_comp.properties` when you modify database properties.

5. Connect to the JCR database by running `db2 connect to <jcrdb> user <userid> using <password>`. You must connect to the JCR database using the userid and password of the WebSphere Portal JCR data source.

6. Execute the following command in the previous substep. If the command completes successfully, the UDF is registered correctly as follows:
   ```
   values schema.sortkeyj('abc','en')
   ```

7. Edit the `icm.properties` file, located in `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm` directory. Add the following section to the end of the file:
   ```
   # Enable/Disable collation support for all DB2 platforms
   ```
6. Enter the following command to start the WebSphere Portal server:

```
./startServer.sh WebSphere_Portal
```

Parent topic: Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Modifying database properties

Next topic: Configure WebSphere Portal to use DB2
Configure WebSphere Portal to use DB2

View the steps to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties

Tips:
- To run these tasks as a non-root user, you must first run the task chown -R non-root_userWebSphereDir.
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

Steps for transferring data to another supported database

1. If you are running a type 2 connection, before transferring data edit the db2cli.ini file. Failure to follow these steps will cause the database transfer to hang at the task action-process-constraints.
   A. Locate the file /home/db2inst1/sqllib/cfg/db2cli.ini.
   B. Add the following to the end of the file. Leave an empty line after ReturnAliases=0.[COMMON]

   ```plaintext
   [COMMON]
   DYNAMIC=1
   ReturnAliases=0
   ```

2. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.

3. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2.
   ```sh
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```

4. Enter the following commands to validate configuration properties...
   ```sh
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```

5. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.
6. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td>./stopServer.sh server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

7. Transfer the database: **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

A. Change to the directory `wp_profile_root/ConfigEngine`.

B. Enter the following command: `./ConfigEngine.sh database-transfer -DWasPassword=password`

**Note:**
- To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command: `./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password`

- If you have been storing data in IBM® Derby for a long time, database transfer could fail with OutOfMemory exceptions. If database transfer fails, add the following property to the command in this step: `./ConfigEngine.sh database-transfer -DDbtJavaMaxMemory=1536M -DWasPassword=password`

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

8. After transferring the database tables, perform a `reorg` check to improve performance. Perform this step for each database alias in the property file.

A. Connect to a database with the following command: `db2 connect to database_alias user db2admin_userid using password`

**Note:** Additional options might be required if additional security has been installed. Refer to DB2 Universal Database commands by example for links to the command reference.

B. After it is connected, run the following commands from the DB2 prompt: `db2 reorgchk update statistics on table all > xyz.out`

C. Look in the reorg column for entries marked with a * (star or asterisk) in the file `xyz.out`. For each line with a *, note the `tablename` and run the following command for each `tablename`: `db2 reorg table tablename db2 terminate db2rbind database_name -l db2rbind.out -u db2admin -p password`

D. The output file `db2rbind.out` is only created when there is an error for the `db2rbind` command.

9. Change to the directory `wp_profile_root/bin`.

10. Enter the following command to start the WebSphere Portal server: `./startServer.sh WebSphere_Portal`
Configuring DB2 for large file handling in Web Content Management

If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

**Prerequisites**
- Installing DB2
- Create users
- Creating remote databases
- Modifying database properties
- Configure WebSphere Portal to use DB2

**Note:** You only need to perform these steps if you are using Web Content Management.

1. Log into the WebSphere Application Server administrative console.
2. Click **Resources** > **JDBC** > **Data sources**.
3. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
4. Select the name of the data source that is defined in `wkplc_comp.properties` for the JCR database domain. The default data source is `wpdbDS`.
5. Click **Custom properties**.
6. Ensure that the `fullyMaterializeLobData` property is set to `false`.

**Parent topic:** Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Configure WebSphere Portal to use DB2

**Next topic:** Changing driver types
Changing driver types

WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Before you begin, ensure that the following conditions are met:
- The WebSphere Portal database has been successfully transferred to DB2 using the `database-transfer` configuration task.
- The files `wkplc_comp.properties` and `wkplc_dbtype.properties` have been modified to set the correct values for the DB2 drivers that you are switching to:

  In the file `wkplc_comp.properties` set each `<Domain>.DbUrl` property using the following formats:
  - `db2 (type 2):        { jdbc:db2:wpsdb }`
  - `db2 (type 4):        { jdbc:db2://<YourDatabaseServer>:50000/wpsdb:returnAlias=0; }

  In the file `wkplc_dbtype.properties` set the `db2.DbLibrary` property using the following format:
  - For DB2 Type 2 driver use `<SQLLIB>/java/db2java.zip`
  - For DB2 Type 4 driver use `<SQLLIB>/java/db2jcc.jar;<SQLLIB>/java/db2jcc_license_cu.jar`

  In the file `wkplc_dbtype.properties` set the `db2.DbDriver` property using the following format:
  - For DB2 Type 2 driver use `COM.ibm.db2.jdbc.app.DB2Driver`
  - For DB2 Type 4 driver use `com.ibm.db2.jcc.DB2Driver`

`Note:` If WebSphere Portal is installed on the same machine as the DB2 server and you switch from a JDBC Type 4 connection to a JDBC Type 2 connection, verify that you have created the alias names for the DB2 databases as described in `Creating remote databases` and that the alias names are specified for the databases in the file `wkplc_comp.properties`. When switching from a JDBC Type 2 connection to a JDBC Type 4 connection, remove the database alias names and refer to the databases directly. This is required because of a limitation in the DB2 Universal JDBC driver.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties
  - Configure WebSphere Portal to use DB2

1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.
2. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2.

   `/home/db2inst1/sqllib/db2profile`

   where `db2inst1` represents your database instance

   `Note:` You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.

3. Enter the following commands to validate configuration properties.

   ```
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,
   ```
From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.

Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td><code>./stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td><code>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

6. Change to the directory `wp_profile_root/ConfigEngine`.

7. To change from one supported driver to the other, run the following task to connect the database, including only the domains that require the switch.

```
```

8. Change to the directory `wp_profile_root/bin`.

9. Enter the following command to start the WebSphere Portal server:

```
./startServer.sh WebSphere_Portal
```

Parent topic: Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Configuring DB2 for large file handling in Web Content Management
Preparing DB2 for z/OS for a stand-alone production server

Setting up the DB2 for z/OS database includes creating user IDs, databases, and table spaces on a remote server.

1. **Installing DB2 for z/OS**
   View information on how to install DB2 for z/OS for use with IBM® WebSphere® Portal.

2. **Creating users**
   Complete this procedure to set up users for IBM DB2 Universal Database™ for z/OS® to work with IBM WebSphere Portal.

3. **Creating remote databases**
   A remote database resides on a different machine than WebSphere Portal. You must manually create the required databases before configuring WebSphere Portal to work with DB2 for z/OS.

4. **Creating the Java Content Repository database**
   View information on setting up your Java Content Repository database to work with WebSphere Portal.

5. **Optional: Assigning custom table spaces**
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

6. **Modifying database properties**
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

7. **Configuring WebSphere Portal to use DB2 for z/OS**
   View information on manually transferring data to the DB2 for z/OS you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository to DB2 for z/OS. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

8. **Verifying database connections**
   After WebSphere Portal is configured to work with your database, test the database connection to ensure that it operates correctly.

**Parent topic:** Configuring databases
Installing DB2 for z/OS

View information on how to install DB2 for z/OS for use with IBM® WebSphere® Portal.

The following prerequisites must exist separately from the WebSphere Portal installation:

- If you are using the DB2 Legacy Type 2 JDBC driver, configure your DB2 Connect client with the following commands:
  - `db2 update dbm cfg using tp_mon_name WAS`
  - `db2 update dbm cfg using spm_name hostname`, where `hostname` is the host name for the machine where the DB2 Connect client is installed (same as portal machine).

- If you are using the DB2 Legacy Type 2 JDBC driver, enable extended shared memory usage with the following commands:
  - `export EXTSHM=ON`
  - `db2set DB2ENVLIST=EXTSHM`
  - `db2start`

  To make the change permanent, you must add the environment variable by adding the following to the `profile.env` file:

  ```
  DB2ENVLIST='EXTSHM'
  ```

  in `/home/db2inst/sqllib/userprofile`

  add: `export EXTSHM=ON`

  **Note:** The shell must be reopened before restarting DB2 for z/OS.

- Both type 2 and type 4 drivers are supported. If you are using the DB2 Legacy Type 2 JDBC driver, the DB2 Connect client must be installed on the same machine as WebSphere Portal to connect to the remote database.

- The DB2 subsystem must be on a supported z/OS platform.

- Update the BP8K0 catalog bufferpool to 35,000 buffers before performing database transfer. You should change this value based on your environment. The SYSIBM.SYSDATABASE table resides in this bufferpool and is used extensively by DB2 for z/OS during database transfer.

To use DB2 for z/OS as the database software for WebSphere Portal, you must have DB2 for z/OS installed on your z/OS system. Refer to the DB2 for z/OS product documentation for instructions. Refer to the Planning for DB2 for z/OS topic for additional considerations for configuring DB2 for z/OS as the WebSphere Portal database.

**Parent topic:** Preparing DB2 for z/OS for a stand-alone production server

**Next topic:** Creating users
Creating users

Complete this procedure to set up users for IBM® DB2 Universal Database™ for z/OS® to work with IBM WebSphere® Portal.

Before you begin: You should have completed Installing DB2 for z/OS.

**- Prerequisites**

- Installing DB2 for z/OS

- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

- If you are configuring multiple WebSphere Portal instances to use a single DB2 for z/OS subsystem, you must create a unique database user for each WebSphere Portal instance. By default all database table names include the name of the database user used to access the data. Therefore, to prevent table name conflicts, you must create a unique database user for each WebSphere Portal instance on the shared DB2 for z/OS subsystem.

- Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Use the following steps to grant access permissions to the database users. Repeat these steps for each WebSphere Portal instance you are setting up.

1. Create all database user IDs in the security product you are using on z/OS. For jcrschema, the database schema name for IBM Java Content Repository data, create a group and connect it to the database user ID for Java Content Repository data, jcr. The following sample shows the RACF definition of such a user ID and group, where jcr is the database user ID for Java Content Repository data, yourDefaultUserGroup is your default RACF group for database user IDs, jcrschema is the database schema name for Java Content Repository data, and yourDefaultGroup is your default RACF group for groups. If you have some other security product such as Top Secret or ACF2 instead of RACF, translate the sample RACF definition into the appropriate syntax before running the job.

```
ADDUSER jcr DFLTGRP(yourDefaultUserGroup) NAME('WAS DB2 ACCESS USER')
PW USER(jcr) NOINTERVAL
ALU jcr PASSWORD(********) NOEXPIRED
ADDGROUP jcrschema SUPGROUP(yourDefaultGroup)
CONNECT jcr GROUP(jcrschema)
```

2. Run the following SQL statements using a tool like SPUFI while logged on to the DB2 subsystem to grant appropriate rights on the newly created databases. If you are configuring multiple WebSphere Portal instances to use a single DB2 for z/OS subsystem, be sure to use the database user associated with the **Note:** Make sure your vmmdb user follows these same standards.

**WebSphere Portal instance you are setting up.**

(C) create/alter tablespaces

(C) create/alter tables

(C) create/alter indice;

(C+R) read/write data

(C) - at configuration time

(R) - at runtime
GRANT USE OF ALL BUFFERPOOLS TO releaseusr;
GRANT USE OF ALL BUFFERPOOLS TO communityusr;
GRANT USE OF ALL BUFFERPOOLS TO customizationusr;

GRANT DBADM ON DATABASE jcrdbnameonzos TO jcr;
GRANT USE OF ALL BUFFERPOOLS TO jcr;
GRANT DBADM ON DATABASE fdbkdbnameonzos TO feedback;
GRANT USE OF ALL BUFFERPOOLS TO feedback;
GRANT DBADM ON DATABASE lmbdbnameonzos TO lmbusr;
GRANT USE OF ALL BUFFERPOOLS TO lmbusr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO releaseusr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO communityusr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO customizationusr;

GRANT SELECT ON SYSIBM.SYSTABLES TO releaseusr;
GRANT SELECT ON SYSIBM.SYSTABLES TO communityusr;
GRANT SELECT ON SYSIBM.SYSTABLES TO customizationusr;

GRANT SELECT ON SYSIBM.SYSCOLUMNS TO jcr;
GRANT SELECT ON SYSIBM.SYSTABLES TO jcr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO fdbkdbusr;
GRANT SELECT ON SYSIBM.SYSTABLES TO fdbkdbusr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO lmbusr;
GRANT SELECT ON SYSIBM.SYSTABLES TO lmbusr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO releaseusr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO communityusr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO customizationusr;

GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO jcr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO fdbkdbusr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO lmbusr;
GRANT SELECT ON SYSIBM.SYSRELS TO releaseusr;
GRANT SELECT ON SYSIBM.SYSRELS TO communityusr;
GRANT SELECT ON SYSIBM.SYSRELS TO customizationusr;

GRANT SELECT ON SYSIBM.SYSRELS TO jcr;
GRANT SELECT ON SYSIBM.SYSRELS TO fdbkdbusr;
GRANT SELECT ON SYSIBM.SYSRELS TO lmbusr;
GRANT USE OF STOGROUP jcrstogroup TO jcr;
GRANT CREATEIN, DROPIN ON SCHEMA jcrschema TO jcr;
GRANT SELECT ON SYSIBM.SYSTABLESPACE TO jcr;
GRANT SELECT ON SYSIBM.SYSVIEWS TO jcr;
GRANT SELECT ON SYSIBM.SYSDUMMY1 TO jcr;
GRANT SELECT ON SYSIBM.SYSTRIGGERS TO jcr;
GRANT SELECT ON SYSIBM.SYSINDEXPART TO jcr;
GRANT SELECT ON SYSIBM.SYSINDEXES TO jcr;
GRANT SELECT ON SYSIBM.SYSSYNONYMS TO jcr;

where:
- releasenameonzos, communitynameonzos, customizationnameonzos, and releaseusr, communityusr, customizationusr represent the databases and database users, respectively, of the WebSphere Portal instance you are setting up. (These users must be created on the host system.)
- jcrdbnameonzos and jcr are the database and database user, respectively, for Content Repository data.
- fdbkdbnameonzos and feedback are the database and database user, respectively, for Feedback data.
- lmdbnameonzos and lmdbusr are the database and database user, respectively, for Likeminds data.
- jcrschema is the database schema name for Content Repository data.

3. Grant the necessary access rights to all users who might require them. Depending on the architecture that you choose, these users might include Java Content Repository and Feedback users.

**Parent topic:** Preparing DB2 for z/OS for a stand-alone production server

**Previous topic:** Installing DB2 for z/OS

**Next topic:** Creating remote databases
Creating remote databases

A remote database resides on a different machine than WebSphere Portal. You must manually create the required databases before configuring WebSphere Portal to work with DB2 for z/OS.

Before you begin, complete the following prerequisites:

- These instructions assume that you will use a local IBM® DB2 Universal Database™ Enterprise Server Edition Connect to connect to a remote IBM DB2 Universal Database for z/OS® server, and that IBM WebSphere® Application Server, WebSphere Portal, and DB2 Connect will be installed on the same machine.
- To run SQL statements, you can use a tool like SPUFI. The specified statements include CREATE STOGROUP, CREATE DATABASE and CREATE TABLESPACE. For CREATE STOGROUP you have to replace the icmvolumes and icmvcat variables with volume serial numbers and a catalog name for your environment. For CREATE TABLESPACE you can specify a specific BUFFERPOOL. Refer to the DB2 for z/OS SQL Reference for a description of these options.
- Ensure that a TEMP database is created for your subsystem. You can use the following statements to create a TEMP database:

  CREATE DATABASE db_name
  AS TEMP;

  CREATE TABLESPACE ts_name
  IN db_name;

- Log on to the DB2 subsystem on the host server. DB2 system administrator rights are needed to create the databases.
- Replace variables as follows:

  - releasenameonzos, communitynameonzos, and customizationnameonzos are the WebSphere Portal databases for WebSphere Portal and Member Manager data.
  - fdbkdbnameonzos and fdbkdbts are the database and table space, respectively, for Feedback data.
  - lmdbnameonzos and lmdbts are the database and table space, respectively, for LikeMinds data.
- Because large objects are stored in columns that can become very large, logging changes to these columns requires a huge amount of log space. For this reason, large object (LOB) logging is disabled by default for tablespaces that contain such data. With LOB logging disabled, you can recover full backups, but not incremental backups that can be used for point in time recovery. To recover point in time backups, you must enable LOB logging. For detailed instructions, see Technote 1306637, Managing LOB logging in DB2 for z/OS.

Prerequisites

- Installing DB2 for z/OS
- Creating users

Use the following steps to set up WebSphere Portal with a remote instance of DB2 for z/OS. Refer to Planning for DB2 for z/OS for a list of databases and table space names. If you are configuring multiple WebSphere Portal instances to use a single DB2 subsystem, be sure to use the database and table space names associated with the WebSphere Portal instance you are setting up.

1. CREATE DATABASE releasenameonzos CCSID UNICODE;
2. CREATE DATABASE communitynameonzos CCSID UNICODE;
3. CREATE DATABASE customizationnameonzos CCSID UNICODE;
4. Execute the steps in the topic Creating the Java Content Repository database.
5. CREATE DATABASE fdbkdbnameonzos CCSID UNICODE;
6. CREATE TABLESPACE fdbkdbts IN fdbkdbnameonzos USING STOGROUP SYSDEFLT PRIQTY 5000 SECQTY 500 LOCKSIZE ROW;
7. CREATE DATABASE lmdbnameonzos CCSID UNICODE;
CREATE TABLESPACE lmdbts IN lmdbnameonzos USING STOGROUP SYSDEFLT PRIQTY 5000 SECQTY 500;

Parent topic: Preparing DB2 for z/OS for a stand-alone production server
Previous topic: Creating users
Next topic: Creating the Java Content Repository database
Creating the Java Content Repository database

View information on setting up your Java Content Repository database to work with WebSphere Portal.

First, you need to create multiple databases for scalability.

The IBM Java Content Repository database grows in size as the IBM® WebSphere® Portal server is used. To support this growth, which includes the dynamic creation of many new tables within the database, the WebSphere Portal server allows you to create multiple databases within a single IBM DB2 Universal Database™ for z/OS® subsystem. A minimum of two databases is recommended, but more can be created if necessary. A single database setup is not recommended; it will quickly become constrained. See the Performance guides on the product documentation Web site for more information.

Each database that is created must begin with a common prefix; there is no convention required for the remainder of the name. For example, to create \textit{xx} number of databases, you may choose to use the following commands:

\begin{verbatim}
CREATE DATABASE JCRDB01
CREATE DATABASE JCRDB02
...
CREATE DATABASE JCRDBxx
\end{verbatim}

In this case, \textit{JCR} is the common prefix. You also have to select a maximum number of user-defined tables (UDT) that each database will be allowed to hold; the recommended value is 400.

- **Prerequisites**
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases

A sample script of the commands that you can use to create the Java Content Repository database in your DB2 for z/OS installation is shown below. The sample demonstrates the creation of a single database. To follow the recommendation of creating at least two databases, duplicate the commands for each additional database as indicated below. Find a template of these commands in the file \texttt{PortalServer_root/installer/wp.config/config/templates/db/db2_zos/jcr_sample.sql}.

**Notes:**
- It is not necessary to duplicate every tablespace definition in every database.
- In order to run the commands shown in the sample, you must know the database name and the IDs of the MVS volumes where the Java Content Repository database tables will be stored.
- Edit the template file for your installation environment before running the commands shown in the sample:
  - Replace \texttt{jcrdbnameX} with the name of the database. Remember that each database name must begin with a common prefix.
  - Replace \texttt{stogroup} with the name of your storage group.
  - Replace \texttt{icmvolumes} with the IDs of the MVS volumes where the Java Content Repository database tables will be stored. These values are assigned by the database administrator.
  - Replace \texttt{icmvcat} with the name of the virtual catalog.
  - Replace \texttt{jcr} with the name of database user ID.
  - Replace \texttt{4kbp} with the name of your 4K bufferpool.
  - Replace \texttt{32kbp} with the name of your 32K bufferpool.
  - Replace \texttt{jcrschema} with the schema name of your Java Content Repository domain.
--DROP DATABASE jcrdbnameX
--DROP STOGROUP stogroup
--DROP ALIAS jcrschema.ICMFICTITIOUS

CREATE STOGROUP stogroup VOLUMES (icmvolumes) VCAT icmvcat

GRANT USE OF STOGROUP stogroup to jcr

CREATE DATABASE jcrdbnameX STOGROUP stogroup BUFFERPOOL 4kbp INDEXBP 4kbp CCSID UNICODE

Note: The following two tablespace commands must be created in every database:

CREATE TABLESPACE ICMLFQ32 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICMTSQ04 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

Note: The following tablespaces are required in the first database only:

CREATE TABLESPACE ICMVFQ04 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICMSFQ04 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSTADO IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRIDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRSCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRISLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRGCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRIGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSTDPV IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSDOAC IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RESMLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COLNLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USERRLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USEGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE SYSCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CACLLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CPRERLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTDLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTGLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NLSKLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE MAXKLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NLSLLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTDLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTGLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NLSKLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE MAXKLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NLSLLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE ICUT301 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TEICLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00208 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00209 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00201 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00202 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00203 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00204 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00205 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00200 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TIELLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE REPLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE REPRLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE RI11LSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE XDOFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE XDOOLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE IDELLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TEICLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE SYAELSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00206 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00207 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00205 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00204 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00203 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00202 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00201 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00200 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE ICMACTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE ICMACCLS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE ICMACLS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE ICUT301 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT311 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT321 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT331 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT341 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT601 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ADDPJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE DWSLJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE GENDJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE GLBPJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE LINKJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE MAXSJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NODDJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NODTJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NODRJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NSPRJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NSURJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRPDJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ROOTJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NODEJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE RWSJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE SPRTJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TIMEJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TSERJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TSINJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
683


LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TSPEJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TSTAJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RIDSJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE WSNOJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 5000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE LOB TABLESPACE ICMLOBT1 IN jcrdbnameX LOG NO
CREATE LOB TABLESPACE ICMLOBT2 IN jcrdbnameX LOG NO
CREATE LOB TABLESPACE ICMLOBT3 IN jcrdbnameX LOG NO
CREATE ALIAS jcrschema.ICMFICTITIOUS FOR SYSIBM.SYSDUMMY1
CREATE TABLESPACE DLKRJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE LKRLJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

**Parent topic:** [Preparing DB2 for z/OS for a stand-alone production server](#)
**Previous topic:** [Creating remote databases](#)
**Next topic:** [Assigning custom table spaces](#)
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:

- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the \wp_profile_root\PortalServer\config\tablespaces\dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.
- When using IBM® DB2 Universal Database™ for z/OS® as a data store, WebSphere Portal requires that its indexes are not padded. Therefore, you must set the DSNZPARM parameters to RETVLCFK=NO or PADIX=NO, or both.

- **Prerequisites**
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases
  - Creating the Java Content Repository database

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file \wp_profile_root\PortalServer\config\tablespaces\dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.index_name.indexspace
   
   For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds

3. Assign a table space to each .tablespace entry in the mapping file. Assignments to .indexspace entries are ignored.
   
   The table space name must be qualified by the database name and prepended by the keyword IN and a space. For example: community.COMP_INST.tablespace=IN COMM8KSPACE

   Repeat this step for each domain that you are transferring.
4. Save and close dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

Parent topic: Preparing DB2 for z/OS for a stand-alone production server
Previous topic: Creating the Java Content Repository database
Next topic: Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- Prerequisites
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases
  - Creating the Java Content Repository database

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

Tip: For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
  - WebSphere Portal 6.1.5 wkplc.properties file reference
  - WebSphere Portal 6.1.5 wkplc_comp.properties file reference
  - WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Working with properties files:
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:
  - release.DbName = release
  - jcr.DbName = jcrdb
  - feedback.DbName = fdbkdb
  - likeminds.DbName = lmdb
  - community.DbName = commdb
  - customization(DbName = custdb

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds
The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:

- `dbdomain.DbType`
- `dbdomain.DbName`
- `dbdomain.DbUrl`
- `dbdomain.DbSchema`

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If `DbUser`, `DbUrl`, and `DbPassword` are not the same across domains, the value for `DataSourceName` must differ from the `DataSourceName` of the other domains. In other words, this value must be unique for the database domain.

**Note:** To successfully transfer data from the JCR domain, you must use the DDF location value for the value of jcr.DbName field when setting up IBM DB2 Universal Database™ for z/OS®. You can locate the name of the DDF location value in the IBM DB2 Universal Database for z/OS sdsnsamp dataset, member DSNTIJUZ, or by running the following DB2 command:

```
db2 subsystem prefix display ddf
```

1. Locate the following files and create a backup copy of each before changing any values:

   - `wp_profile_root/ConfigEngine/properties/wkplc.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_comp.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties`

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In `wkplc_comp.properties`, most properties are repeated for each domain.

2. Use a text editor to open the properties file `wkplc_comp.properties` and modify the values to correspond to your environment.

   A. For `dbdomain.DbType`, type `db2_zos`.
   B. For `dbdomain.DbName`, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the `dbdomain.DbUrl` property.
   C. For `dbdomain.DbSchema`, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   D. For `dbdomain.DbNameOnZos`, type the name of the WebSphere Portal database on DB2 for z/OS. **Note:**
      - If running DB2 for z/OS as a remote database, set the value to the name of the remote database for the domain.
      - If WebSphere Portal is running on z/OS with DB2 for z/OS, set the value equal to the value of DbName.
   E. For `dbdomain.DataSourceName`, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - `releaseDS`
      - `communityDS`
      - `customizationDS`
      - `jcrDS`
      - `lmdbDS`
      - `feedback`
   F. For `dbdomain.DbUrl`, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:** The database element of this value should match the value of DbName.
G. **For dbdomain.DbUser**, type the user ID for the database administrator.

H. **For dbdomain.DbPassword**, type the password for the database administrator.

I. **For dbdomain.DbTablespace**, type the name of the DB2 for z/OS tablespace.

J. **For dbdomain.DbStorageGroup**, type the name of the storage group for the database.

K. **For dbdomain.DbVolumes**, type the volumes for the database.

L. **For dbdomain.DbVcatal**, type the VCAT for the database.

M. **For dbdomain.Db4KBufferPoolName**, type the 4K bufferpool name for the database.

N. **For dbdomain.Db32KBufferPoolName**, type the 32K bufferpool name for the database.

O. **For dbdomain.DbIndex4KBufferPoolName**, type the 4K bufferpool name for the database. If you choose to use the default bufferpool value BP3, verify that this bufferpool is active.

P. **For dbdomain.TablespaceTrackMod**, set the value to determine TRACKMOD attribute of all tablespaces to use the specified value. Refer to the DB2 for z/OS documentation before changing this value.

3. **(Optional)** When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

   A. **For source.domainDbType**, type of the database you are currently configured to use. The value for source.domainDbType is Derby by default.

   B. **For source.domainDbName**, type the name of the database domain you are currently using.

   C. **For source.domainDbSchema**, type current schema identifier for objects within the database for this domain.

   D. **For source.domainDataSourceName**, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

   E. **For source.domainDbUrl**, type the url currently used to access your database.

   F. **For source.domainDbUser**, type the name of the user accessing this database.

   G. **For source.domainDbPassword**, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file wkplc_dbtype.properties.

   A. **For db2_zos.DbDriver**, type the name of the class that SqlProcessor uses to import SQL files.

   B. **For db2_zos.DbLibrary**, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. **For db2_zos.JdbcProviderName**, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

   D. **For db2_zos.DbDriverType**, type the number of the driver type for the database.

   E. **For db2_zos.DbLocationName**, type the DB2 location name. This value is set in the installation job DSNTIJUZ.

6. Save and close the file.

7. Update the following property in the file wkplc.properties.

   A. **For WasPassword**, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Preparing DB2 for z/OS for a stand-alone production server

**Previous topic:** Assigning custom table spaces

**Next topic:** Configuring WebSphere Portal to use DB2 for z/OS
Configuring WebSphere Portal to use DB2 for z/OS

View information on manually transferring data to the DB2 for z/OS you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository to DB2 for z/OS. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

- Prerequisites
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases
  - Creating the Java Content Repository database
  - Modifying database properties

Steps for transferring data to another supported database

1. If you are running a type 2 connection, before transferring data edit the db2cli.ini file. Failure to follow these steps will cause the database transfer to hang at the task action-process-constraints.
   A. Locate the file /home/db2inst1/sqllib/cfg/db2cli.ini.
   B. Add the following to the end of the file. Leave an empty line after ReturnAliases=0.[COMMON]
      DYNAMIC=1
      ReturnAliases=0

2. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.

3. Enter the following commands to validate configuration properties...

4. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td>./stopServer.sh server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

6. Transfer the database:

   **Note**: Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.
   A. Change to the directory wp_profile_root/ConfigEngine.
B. Enter the following command: 

```
./ConfigEngine.sh database-transfer -DWasPassword=password
```

**Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command: 

```
./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
```

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

7. From the command line processor, reset the check pending status on the WebSphere Portal table spaces listed here.

CHECK DATA is a DB2 batch utility that is invoked through JCL or through the DB2 utility panels. Type the following utility commands to check pending status:

```
check data tablespace releasenameonzos.TS280A
check data tablespace releasenameonzos.TS300A
check data tablespace releasenameonzos.TS2110A
check data tablespace releasenameonzos.TS830A
check data tablespace communitynameonzos.TS8000B
check data tablespace communitynameonzos.TS8011B
check data tablespace communitynameonzos.TS280B
check data tablespace customizationnameonzos.TS2110C
check data tablespace jcrdbnameonzos.ICMSFQ04
check data tablespace jcrdbnameonzos.TS2110D
```

where `releasenameonzos`, `communitynameonzos`, and `customizationnameonzos` are the names of your WebSphere Portal databases, and `jcrdbnameonzos` is the name of your JCR database. Refer to the Utility Guide and Reference for DB2 for z/OS for additional details.

8. Run the `RUNSTATS` utility as shown in the following example:

```
LISTDEF JCRDBZOS INCLUDE TABLESPACE JCRDBZOS.* BASE
RUNSTATS TABLESPACE LIST JCRDBZOS INDEX(ALL) KEYCARD TABLE(ALL)
LISTDEF JCRDB01 INCLUDE TABLESPACE JCRDB01.* BASE
RUNSTATS TABLESPACE LIST JCRDB01 INDEX(ALL) KEYCARD TABLE(ALL)
...
```

9. Start the WebSphere Application Server.

10. Verify that the WebSphere Portal application server is running by opening the following URL in a browser:

```
http://hostname.companyname.com:port_number/wps/portal
```

where `hostname.companyname.com` is the fully qualified host name of the machine where WebSphere Portal is running and `port_number` is the transport port that is created by WebSphere Application Server.

If you have performed database transfer in a clustered environment, you must explicitly copy the `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm/icm.properties` file from the primary node on which the `database-transfer` task was processed to all secondary nodes. This ensures that the secondary nodes have the new JCR database values specified in the `icm.properties` file.

1. Stop the portal server on the secondary node.
2. Copy the `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm/icm.properties` file from the primary node and replace the `icm.properties` file on the secondary node with the new file from the primary node.
3. Start the portal server on the secondary node.

**Parent topic:** Preparing DB2 for z/OS for a stand-alone production server

**Previous topic:** Modifying database properties

692
Next topic: Verifying database connections
Verifying database connections

After WebSphere Portal is configured to work with your database, test the database connection to ensure that it operates correctly.

- **Prerequisites**
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases
  - Creating the Java Content Repository database
  - Modifying database properties
  - Configuring WebSphere Portal to use DB2 for z/OS

You can verify the connection from a browser or from a command line. To verify that WebSphere Portal is running from a browser, open the portal in a Web browser: http://hostname.yourco.com:port_number/wps/portal, where hostname.yourco.com is the fully qualified host name of the machine where WebSphere Portal is running and port_number is the transport port that is created by IBM® WebSphere® Application Server.

There may be an error if any of the following conditions appear.

- When trying to access the portal you get a 503 error.
- If you had any locale problems with your database, you could see invalid characters, such as ?????, after logging in. This may happen if the character set of the database is not UTF-8 compliant.
- If something went wrong with the data that was transferred, you may not be able to login. WebSphere Portal will indicate you entered an invalid user ID and password even though you know it is valid.

Verify the connection from a command line by completing the following steps:

1. Start a 5250 session on the local machine where WebSphere Portal is installed.
2. For WebSphere Portal on WebSphere Application Server (UserData path), enter the following on the command line: cd wp_profile_root/ConfigEngine.
3. Enter the following command: ConfigEngine.sh validate-database-connection -DTransferDomainList="release,community,customization,jcr,feedback,likeminds" -DWasPassword= password

For security reasons, you should not leave passwords in the wkplc_comp.properties file. Edit the file prior to running a configuration task and insert the passwords that are needed for that task. After the task has run, delete all passwords from the file.

Alternatively, you can specify the password on the command line rather than update the wkplc_comp.properties file. For example: ConfigEngine.sh -DPortalAdminPwd=password -DWasPassword=password validate-wps-admin-login

When installing WebSphere Portal, the passwords in the wkplc_comp.properties file are automatically removed after configuration.

**Parent topic:** Preparing DB2 for z/OS for a stand-alone production server  
**Previous topic:** Configuring WebSphere Portal to use DB2 for z/OS
Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Setting up the Oracle database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

1. Installing Oracle
   View information on how to install Oracle for use with WebSphere Portal.

2. Creating databases
   View some important considerations before setting up Oracle databases to work with WebSphere Portal.

3. Creating users
   View the steps to set up users for Oracle to work with WebSphere Portal.

4. Modifying database properties
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

6. Linux stand-alone server: Creating JCR table spaces
   This topic provides manual instructions for creating JCR table spaces for Oracle.

7. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

8. Configure WebSphere Portal to use Oracle
   This section provides information on how to manually transfer data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configuring databases
Installing Oracle

View information on how to install Oracle for use with WebSphere Portal.
You can use Oracle as the database software. You must install Oracle separately and before performing a database transfer.
Refer to Oracle product documentation for installation instructions.

Parent topic: Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server
Next topic: Creating databases
Creating databases

View some important considerations before setting up Oracle databases to work with WebSphere Portal. For information about creating databases, refer to the Oracle product documentation. For information on the recommended database architecture and the databases you will need to create, see the Planning for Oracle topic. Be sure that all databases to be used with WebSphere Portal are created as UNICODE character set databases.

If you are using remote Oracle 9 or 10 databases, you must also copy the ojdbc14.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM® WebSphere® Application Server Version 7.0, you must also copy the ojdbc6.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM WebSphere Application Server Version 6.1, you must also copy the ojdbc5.jar file from the remote Oracle server to the WebSphere Portal machine. The typical location is the oracle_home/sqldeveloper/jdbc/lib directory. Record the copy location on your local machine for future reference.

When creating Oracle databases for use with WebSphere Portal, you should consider the following information:

- The Oracle databases must be created manually before configuring WebSphere Portal.
- All databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle server will be remote or local to the WebSphere Portal installation.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle database in order for WebSphere Portal to communicate with the Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes:
  - db_block_size = 8192 bytes
  - db_cache_size = 307,200 bytes
  - db_files = 1024 files
  - log_buffer = 65536 bytes
  - open_cursors = 1500 cursors
  - pga_aggregate_target = 204,800 bytes
  - pre_page_sga = true
  - processes = 300 processes
  - shared_pool_size = 204,800 bytes

  **Note:** If you are using IBM Java Content Repository, the open_cursors value may need to be increased based on the table count in the Java Content Repository schema.

- Raise the number of parallel servers as appropriate. For example, if you have more than 875 parallel servers, you should set the parallel_max_servers to 1200.
- The Oracle parameter CURSOR_SHARING allows similar SQL Statements to be shared when possible, which prevents parsing and establishing a new execution plan. The execution plan is used by Oracle to gather the data needed to satisfy a request. There are two options for CURSOR_SHARING, which are as follows:
  - **FORCE**
    - When you select this option, Oracle uses the same execution plan for all SQLs that are similar in value even if the values are different. When you use this option, the execution plan may not provide optimum performance. For example, similar SQLs with different values may behave differently when executed running the same plan.
  - **EXACT**
When you select this option, Oracle only shares the same execution plan for SQLs that are identical and use the same values. This option removes the risk of a SQL statement being executed when optimum performance conditions do not exist.

WebSphere Portal supports both options. Regardless of the option selected, portlet applications should not be affected. Contact your database administrator for further assistance on these options.

**Prerequisites**

- Installing Oracle

**Parent topic:** Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Installing Oracle

**Next topic:** Creating users
Creating users

View the steps to set up users for Oracle to work with WebSphere Portal.

Before you begin:

- You should have installed Oracle.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

**Prerequisites**

- Installing Oracle
- Creating databases

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Start the Oracle SQL command line processor ‘SQL*Plus’ by entering `sqlplus /nolog` on the operating system command prompt

2. Log in to the Oracle database by executing command `connect user/password@database name`. For example: `system/manager@wpsdb` will log the administrative user system with a password of manager into the wpsdb database.

3. Create the WebSphere Portal user `dbdomain.DbUser`, where `dbdomain` is replaced by release, community, or customization.

   A. Log in to the database in which you want to create the new users.

   B. Create the users that you need in the appropriate databases by entering the following commands in the SQL*Plus tool, where `user_tablespace` is the default tablespace that is identified by the database administrator to store user objects and `temp_tablespace` is identified to store temporary objects. All names must be in lower case.

```
SQL> create user releaseusr identified by password
default tablespace user_tablespace
temporary tablespace temp_tablespace;
GRANT UNLIMITED TABLESPACE TO releaseusr;
```

```
SQL> create user communityusr identified by password
default tablespace user_tablespace temporary tablespace temp_tablespace;
GRANT UNLIMITED TABLESPACE TO communityusr;
```

```
SQL> create user customizationusr identified by password
default tablespace user_tablespace
temporary tablespace temp_tablespace;
GRANT UNLIMITED TABLESPACE TO customizationusr;
```

```
SQL> create user jcr identified by password
default tablespace user_tablespace
temporary tablespace temp_tablespace;
GRANT UNLIMITED TABLESPACE TO jcr
```

4. Connect to the Feedback database:

   A. Enter the following command: `SQL> connect`
B. Enter `user-name: username/password@dbname` where `username` is an existing administrative user in the database. For example, `system/manager@fdbkdb` will log the administrative user system with a password of manager into the fdbkdb database.

C. Create the Feedback user:

```
SQL> create user feedback identified by password

default tablespace user_tablespace
temporary tablespace temp_tablespace;
GRANT UNLIMITED TABLESPACE TO feedback
```

D. Log out of the command line tool using the command `SQL> exit`.

5. Connect to the LikeMinds database:

A. Enter the following command:

```
SQL> connect
```

B. Enter `user-name: username/password@dbname`, where `username` is an existing administrative user in the database.

For example, `system/manager@lmdb` will log the administrative user system with a password of manager into the lmdb database.

C. Create the LikeMinds user:

```
SQL> create user lmdbusr identified by password

default tablespace user_tablespace
temporary tablespace temp_tablespace;
GRANT UNLIMITED TABLESPACE TO lmdbusr
```

D. Log out of the command line tool using the command `SQL> exit`.

6. Perform the following steps to verify that you can connect to the database with the user ID and password that you just created:

A. Enter the `SQL> connect` command to connect to the content database.

B. Enter `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example, `system/manager@jcrdb` will log the administrative user system with a password of manager into the Oracle database.
Modifying database properties
This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Installing Oracle
  - Creating databases
  - Creating users

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:
  - release.DbName = release
  - jcr.DbName = jrcdb
  - feedback.DbName = fdbkdb
  - likeminds.DbName = lmdb
  - community.DbName = commdb
  - customization.DbName = custdb
- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
- `dbdomain.DbType`
- `dbdomain.DbName`
- `dbdomain.DbUrl`
- `dbdomain.DbSchema`

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If `DbUser`, `DbUrl`, and `DbPassword` are not the same across domains, the value for `DataSourceName` must differ from the `DataSourceName` of the other domains. In other words, this value must be unique for the database domain.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

1. Locate the following files and create a backup copy of each before changing any values:
   - `<wp_profile_root>/ConfigEngine/properties/wkplc.properties`
   - `<wp_profile_root>/ConfigEngine/properties/wkplc_comp.properties`
   - `<wp_profile_root>/ConfigEngine/properties/wkplc_dbtype.properties`

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In `wkplc_comp.properties`, most properties are repeated for each domain.

2. Use a text editor to open the properties file `wkplc_comp.properties` and modify the values to correspond to your environment.
   A. For `dbdomain DbType`, type `oracle`.
   B. For `dbdomain DbName`, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the `dbdomain DbUrl` property.
   C. For `dbdomain DbSchema`, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The `dbdomain DbName` should be the same value used for the `dbdomain DbSchema`.
      
      **Restriction:** The value for `dbdomain DbSchema` must equal the value for `dbdomain DbUser`.
   D. For `dbdomain DataSourceName`, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - `releaseDS`
      - `communityDS`
      - `customizationDS`
      - `jcrDS`
      - `lmdbDS`
      - `feedback`
   E. For `dbdomain DbUrl`, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:**
      - The database element of this value should match the value of `DbName`.
      - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following: `jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME`. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.
F. For `dbdomain.DbUser`, type the user ID for the database administrator. **Restriction:** The value for `dbdomain.DbUser` must equal the value for `dbdomain.DbSchema`.

G. For `dbdomain.DbPassword`, type the password for the database administrator.

H. For `dbdomain.DBA.DbUser`, type the database administrator user ID for privileged access operations during creation of the database.

I. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.

J. For `dbdomain.DbHome`, type the root location for the database. **Note:** This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

   A. For `source.domainDbType`, type of the database you are currently configured to use. The value for `source.domainDbType` is Derby by default.

   B. For `source.domainDbName`, type the name of the database domain you are currently using.

   C. For `source.domainDbSchema`, type current schema identifier for objects within the database for this domain.

   D. For `source.domainDataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

   E. For `source.domainDbUrl`, type the url currently used to access your database.

   F. For `source.domainDbUser`, type the name of the user accessing this database.

   G. For `source.domainDbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

   A. For `oracle.DbDriver`, type the name of the class that `SqlProcessor` uses to import SQL files.

   B. For `oracle.DbLibrary`, type the directory and name of the `.zip` or `.jar` file that contains the JDBC driver class.

   C. For `oracle.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

---

**Parent topic:** Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Creating users

**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

- **Prerequisites**
  - Installing Oracle
  - Creating databases
  - Creating users
  - Modifying database properties

1. On the database server, make sure that the folder `db_1` contains the subfolders `your_oracle_instance/data` and `your_oracle_instance/index`. If this folder hierarchy does not exist, create it manually before you run the `setup-database` task. The `setup-database` task requires these folders to create database users. If these folders do not exist, the `setup-database` task will fail.

2. Change to the directory `wp_profile_root/ConfigEngine`

3. To create the database users, type the following command:

   ```bash
   ./ConfigEngine.sh setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:

   `<WP_root>/base/wp.db.impl/config/templates/setupdb/sqlserver2005/<database domain>/createUsers.sql`

---

**Parent topic:** Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Modifying database properties

**Next topic:** Linux stand-alone server: Creating JCR table spaces

**Related tasks**

Manually creating users and granting privileges for Oracle
Linux stand-alone server: Creating JCR table spaces

This topic provides manual instructions for creating JCR table spaces for Oracle.

- Prerequisites
  - Installing Oracle
  - Creating databases
  - Creating users
  - Modifying database properties
  - Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.
2. Create tablespaces using the following commands as examples:
   Substitute the values of your environment for the following variables:
   - `&jcrdb` is the name of the database you created to store user data.
   - `&dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.

   Ensure that the `.` is included in the variables when you substitute the values of your environment with these variables.

   **Important:** You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

   ```
   create tablespace ICMLFQ32 datafile '&dbpath./&jcrdb./data/ICMLFQ32_01.dbf' size 300M reuse autoextend on maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLNF32 datafile '&dbpath./&jcrdb./data/ICMLNF32_01.dbf' size 25M reuse autoextend on maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLFQ04 datafile '&dbpath./&jcrdb./data/ICMLFQ04_01.dbf' size 25M reuse autoextend on maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMSFQ04 datafile '&dbpath./&jcrdb./data/ICMSFQ04_01.dbf' size 150M reuse autoextend on maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLSNDX datafile '&dbpath./&jcrdb./index/ICMLSNDX_01.dbf' size 10M reuse autoextend on maxsize UNLIMITED extent management local autoallocate;
   ```

   A. Set the `size`, `autoextend`, and `maxsize` values according to your environment. For example, you may want to change the `maxsize` to a set value rather than `UNLIMITED`.
   B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.
   C. Refer to the Oracle command reference for more information about using the `create tablespace` command.

Parent topic: Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Setting up databases

Next topic: Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

- **Prerequisites**
  - Installing Oracle
  - Creating databases
  - Creating users
  - Modifying database properties
  - Setting up databases
  - Linux stand-alone server: Creating JCR table spaces

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`
   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - `release`
   - `community`
   - `customization`
   - `jcr`
   - `feedback`
   - `likeminds`
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword `TABLESPACE` and a space. For example: `community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE`
   Repeat this step for each domain that you are transferring.
4. Save and close `dbdomain.space_mapping.properties`.
5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - `Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
- UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

**Parent topic:** Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server  
**Previous topic:** Linux stand-alone server: Creating JCR table spaces  
**Next topic:** Configure WebSphere Portal to use Oracle
Configure WebSphere Portal to use Oracle

This section provides information on how to manually transfer data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- Prerequisites
  - Installing Oracle
  - Creating databases
  - Creating users
  - Modifying database properties
  - Setting up databases
  - Linux stand-alone server: Creating JCR table spaces

Tips:
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

Steps for transferring data to another supported database

1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.
2. Enter the following commands to validate configuration properties:
   ```
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```
   ```
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```
3. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.
4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td><code>./stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>
5. Transfer the database. **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.
   
   A. Change to the directory `wp_profile_root/ConfigEngine`.
   
   B. Enter the following command:
   
   ```
   ./ConfigEngine.sh database-transfer -DWasPassword=password
   
   **Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:
   
   ```
   ./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
   ```
   
   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.
   
   If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.
   
6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example:
   
   ```
   SQL> execute dbms_stats.gather_schema_stats(ownname=> 'jcr', cascade=> TRUE);
   ```
   
7. Change to the directory `wp_profile_root/bin`.
   
8. Enter the following command to start the WebSphere Portal server:
   
   ```
   ./.startServer.sh WebSphere_Portal
   
   Parent topic: Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server
   
   Previous topic: Assigning custom table spaces
Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Setting up the Oracle RAC database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

1. **Installing Oracle RAC**
   View information on how to install Oracle RAC for use with WebSphere Portal.

2. **Create users**
   View the steps to set up users for Oracle RAC to work with WebSphere Portal.

3. **Creating databases**
   View some important considerations before setting up Oracle RAC databases to work with WebSphere Portal.

4. **Modifying database properties**
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. **Setting up databases**
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

6. **Linux stand-alone server: Creating JCR table spaces on Oracle RAC**
   This topic provides manual instructions for creating JCR table spaces for Oracle.

7. **Optional: Assigning custom table spaces**
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

8. **Configuring WebSphere Portal to use Oracle RAC**
   This section provides information on how to manually transfer data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

**Parent topic:** Configuring databases
Installing Oracle RAC

View information on how to install Oracle RAC for use with WebSphere Portal.

Before you begin:
- You should have completed reviewing the Planning for Oracle RAC section.
- Ensure the Oracle CRS (Cluster Ready Service) and Oracle RAC databases have been installed and configured on the primary and secondary nodes.
- Run the following commands on both RAC nodes to start global services daemon_(GSD), oracle listeners, and agents.
  - `$ gsdctl start`
  - `$ lsnrctl start`
  - `$ agentctl start`

You can use Oracle RAC as the database software for WebSphere Portal. You must install Oracle RAC separately and before performing WebSphere Portal database transfer.

Refer to Oracle product documentation for installation instructions.

**Parent topic:** Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Next topic:** Create users
Create users

View the steps to set up users for Oracle RAC to work with WebSphere Portal.

Before you begin:

- You should have completed installing Oracle RAC.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

- Prerequisites
  - Installing Oracle RAC

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Create the users in the WebSphere Portal databases
   A. Log in to the database in which you want to create the new users.
   B. Create the users that you need in the appropriate databases by entering the following command in the SQL*Plus tool, where user_tablespace is the default tablespace that is identified by the database administrator to store user objects and temp_tablespace is identified to store temporary objects. Tip: Balance the storage of user objects among tablespaces to prevent running out of space with overuse of user_tablespace. Also consider increasing the size of user_tablespace when handling a high volume of users.

   SQL> create user username identified by password
default tablespace user_tablespace
temporary tablespace temp_tablespace;

2. Log in by entering the command $sqlplus in SQL*Plus:

3. Enter the command user-name: username/password@dbname, where username is an existing administrative user in the database. For example: system/manager@wpsdb will log the administrative user system with a password of manager into the wpsdb database.

4. Create the WebSphere Portal user dbdomain.DbUser, where dbdomain is replaced by release, community, customization, or JCR. All names must be in lower case. The following code uses the default user tablespace and the default temporary tablespace. Explicitly specify these tablespaces if you do not want to use the defaults.

   SQL> create user releaseusr identified by password;
   SQL> create user communityusr identified by password;
   SQL> create user customizationusr identified by password;
   SQL> create user jcr identified by password;

5. Connect to the content database by entering the command SQL> connect.

6. Enter the following, where username is an existing administrative user in the database. For example, system/manager@jcrdb will log the administrative user system with a password of manager into the Oracle RAC database.

   SQL> create user jcr identified by password;

7. Create the Oracle RAC users by entering the following: When creating the jcr user, grant all necessary privileges. If you do not grant privileges, you will receive the error ICMADMIN lacks CREATE SESSION Privilege logon denied
when you try to connect with the `jcr` user:

```sql
SQL> create user jcr identified by password default tablespace users temporary tablespace temp;
```

8. **Connect to the Feedback database:**
   A. Enter the following command: `SQL> connect`  
   B. Enter `user-name: username/password@dbname` where `username` is an existing administrative user in the database. For example, `system/manager@fdbkdb` will log the administrative user system with a password of manager into the `fdbkdb` database.

9. **Create the Feedback user:**
   ```sql
   SQL> create user feedback identified by password default tablespace users temporary tablespace temp;
   ```

10. **Connect to the LikeMinds database:**
    ```sql
    SQL> connect
    ```

11. Enter `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example, `system/manager@lmdb` will log the administrative user system with a password of manager into the `lmdb` database.

12. **Create the LikeMinds user:**
    ```sql
    SQL> create user lmdbusr identified by password default tablespace users temporary tablespace temp;
    ```

13. Log out of the command line tool using the command `SQL> exit`.

**Parent topic:** Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server  
**Previous topic:** Installing Oracle RAC  
**Next topic:** Creating databases
Creating databases

View some important considerations before setting up Oracle RAC databases to work with WebSphere Portal. When creating Oracle RAC databases for use with WebSphere Portal, you should consider the following information:

- The Oracle RAC databases must be created manually before configuring WebSphere Portal.

- All databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.

- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.

- Determine if your Oracle RAC server will be remote or local to the WebSphere Portal installation.

- If using an earlier version of Oracle RAC (9i or earlier), ensure that Oracle RAC JVM is also installed.

- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle RAC database in order for WebSphere Portal to communicate with the IBM Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes are:

  db_block_size = 8192 bytes
  db_cache_size = 314,572,800 bytes
  db_files = 1024 files
  log_buffer = 65536 bytes
  open_cursors = 1500 cursors
  pga_aggregate_target = 209,715,200 bytes
  pre_page_sga = true
  processes = 300 processes
  shared_pool_size = 209,715,200 bytes

  **Note:** If you are using IBM Java Content Repository, the open_cursors value may need to be increased based on the table count in the IBM Java Content Repository schema.

- **Prerequisites**
  - Installing Oracle RAC
  - Create users

Refer to the following instructions to create tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.

2. Create the tablespaces. **Important:** You must use the same tablespace names listed in the commands. The tablespace names cannot be customized or modified.

   A. Find and edit the SQL script `jcr_ora_tablespaces.sql` in the directory `wp_profile_root/ConfigEngine/work/db/oracle`.

   B. In the define section, replace the following variables with the values from your environment:

      - `jcrdb`
        - Name of the database you created to store user data.

      - `logfile`
        - Location to store the log file.

      - `dbpath`
        - Directory where you created the database.

   C. Set the size, autoextend, and maxsize values according to your environment. For example, you may want to change the maxsize to a set value rather than UNLIMITED. Consult your Database Administrator for more info.
D. Execute the following SQL script:

Notes:
- The DBA or a user with sufficient credentials (CREATE tablespace) must execute the script.
- The script will prompt for the database username and password.

```sql
# sqlplus
SQL > @jcr_ora_tablespaces.sql
```

Parent topic: Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server
Previous topic: Create users
Next topic: Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Installing Oracle RAC
  - Create users
  - Creating databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:

- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**

- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:
  - `release.DbName` = release
  - `jcr.DbName` = jcrdb
  - `feedback.DbName` = fdbkdb
  - `likeminds.DbName` = lmdb
  - `community.DbName` = commdb
  - `customization.DbName` = custdb

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace `dbdomain` with:
  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`

The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.
   A. For dbdomain DbType, type oracle.
   B. For dbdomain DbName, type the name of the WebSphere Portal domain database. Note: This value is also the database element in the dbdomain.DbUrl property.
   C. For dbdomain DbSchema, type the schema name of the database domain. Note: Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The dbdomain.DbName should be the same value used for the dbdomain.DbSchema

   Restriction: The value for dbdomain.DbSchema must equal the value for dbdomain.DbUser.
   D. For dbdomain DataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For dbdomain DbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. Note:
      - The database element of this value should match the value of DbName.
      - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following:
jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.
F. For dbdomain.DbUser, type the user ID for the database administrator. **Restriction:** The value for dbdomain.DbUser must equal the value for dbdomain.DbSchema.

G. For dbdomain.DbPassword, type the password for the database administrator.

H. For dbdomain.DBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.

I. For dbdomain.DBA.DbPassword, type the database administrator password for privileged access operations during creation of the database.

J. For dbdomain.DbHome, type the root location for the database. **Note:** This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For source.domainDbType, type of the database you are currently configured to use. The value for source.domainDbType is Derby by default.

B. For source.domainDbName, type the name of the database domain you are currently using.

C. For source.domainDbSchema, type current schema identifier for objects within the database for this domain.

D. For source.domainDataSourceName, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For source.domainDbUrl, type the url currently used to access your database.

F. For source.domainDbUser, type the name of the user accessing this database.

G. For source.domainDbPassword, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file wkplc_dbtype.properties.

A. For oracle.DbDriver, type the name of the class that SqlProcessor uses to import SQL files.

B. For oracle.DbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

C. For oracle.JdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file wkplc.properties.

A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Creating databases

**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

Before you begin, ensure that the following prerequisites are met:
- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

- Prerequisites
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties

1. On the database server, make sure that the folder `db_1` contains the subfolders `your_oracle_instance/data` and `your_oracle_instance/index`. If this folder hierarchy does not exist, create it manually before you run the setup-database task. The `setup-database` task requires these folders to create database users. If these folders do not exist, the `setup-database` task will fail.

2. Change to the directory `wp_profile_root/ConfigEngine`

3. To create the database users, type the following command:

   ```bash
   ./ConfigEngine.sh setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:

   `<WP_root>/base/wp.db.impl/config/templates/setupdb/sqlserver2005\<database domain>\createUsers.sql`.

Parent topic: Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Modifying database properties

Next topic: Linux stand-alone server: Creating JCR table spaces on Oracle RAC

Related tasks
Manually creating users and granting privileges for Oracle
Linux stand-alone server: Creating JCR table spaces on Oracle RAC

This topic provides manual instructions for creating JCR table spaces for Oracle.

**Prerequisites**
- Installing Oracle RAC
- Create users
- Creating databases
- Modifying database properties
- Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.
2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:
   - `&jcrdb` is the name of the database you created to store user data.
   - `&dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.

   Ensure that the `;` is included in the variables when you substitute the values of your environment with these variables.
   **Important:** You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

   ```sql
   create tablespace ICMLFQ32 
       datafile '&dbpath./&jcrdb./data/ICMLFQ32_01.dbf' 
       size 300M reuse autoextend on 
       next 10M maxsize UNLIMITED extent management local autoallocate;
   
   create tablespace ICMLNF32 
       datafile '&dbpath./&jcrdb./data/ICMLNF32_01.dbf' 
       size 25M reuse autoextend on 
       next 10M maxsize UNLIMITED extent management local autoallocate;
   
   create tablespace ICMVFQ04 
       datafile '&dbpath./&jcrdb./data/ICMVFQ04_01.dbf' 
       size 25M reuse autoextend on 
       next 10M maxsize UNLIMITED extent management local autoallocate;
   
   create tablespace ICMSFQ04 
       datafile '&dbpath./&jcrdb./data/ICMSFQ04_01.dbf' 
       size 150M reuse autoextend on 
       next 10M maxsize UNLIMITED extent management local autoallocate;
   
   create tablespace ICMLSNDX 
       datafile '&dbpath./&jcrdb./index/ICMLSNDX_01.dbf' 
       size 10M reuse autoextend on 
       next 10M maxsize UNLIMITED extent management local autoallocate;
   ```

   A. Set the `size`, `autoextend`, and `maxsize` values according to your environment. For example, you may want to change the `maxsize` to a set value rather than `UNLIMITED`.

   B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.

   C. Refer to the Oracle command reference for more information about using the `create tablespaces` command.

**Parent topic:** Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Setting up databases

**Next topic:** Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root/PortalServer/config/tables/tables.dbdomain.space_mapping.properties` file.
- For details on creating table spaces refer to the documentation for the database.

Prerequisites
- Installing Oracle RAC
- Create users
- Creating databases
- Modifying database properties
- Setting up databases
- Linux stand-alone server: Creating JCR table spaces on Oracle RAC

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root/PortalServer/config/tables/tables.dbdomain.space_mapping.properties` that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`
   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
     - release
     - community
     - customization
     - jcr
     - feedback
     - likeminds
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword `TABLESPACE` and a space. For example: `community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE`
   Repeat this step for each domain that you are transferring.
4. Save and close `dbdomain.space_mapping.properties`.
5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - Windows: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
- UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

**Parent topic:** Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Linux stand-alone server: Creating JCR table spaces on Oracle RAC

**Next topic:** Configuring WebSphere Portal to use Oracle RAC
Configuring WebSphere Portal to use Oracle RAC

This section provides information on how to manually transfer data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- Prerequisites
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties
  - Setting up databases
  - Linux stand-alone server: Creating JCR table spaces on Oracle RAC

Tips:

- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

- The WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following: jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.

- Manually specify an RAC datasource URL in the WebSphere Application Server console. The URL should look like the following:
  
  jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICE)).

- When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

Steps for transferring data to another supported database
1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.
2. Enter the following commands to validate configuration properties:
   
   ```
   ./ConfigEngine.sh validate-database-driver -DTTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```

3. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.
4. Stop both WebSphere Application Server and the WebSphere Portal server:

   ```
   Option | Description
   ------|--------------------------------------------------------
   WebSphere Application Server | `./stopServer.sh server1 -username admin_userid -password admin_password`
   WebSphere Portal | `./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password`
   ```

5. Transfer the database:

   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory `wp_profile_root/ConfigEngine`.
   B. Enter the following command:
   ```
   ./ConfigEngine.sh database-transfer -DWasPassword=password
   ```

   **Note:** To select specific database domains to transfer, modify the `-DTTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:
   ```
   ./ConfigEngine.sh database-transfer -DTTransferDomainList=jcr -DWasPassword=password
   ```

   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

   If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example:
   ```
   SQL> execute dbms_stats.gather_schema_stats(ownname=>'jcr', cascade=> TRUE);
   ```

7. Specify the JDBC URL to connect to the cluster:

   A. Login to the WebSphere Application Server Administrator Console
   B. Navigate to Resources > JDBC Providers
   C. If there is a value in the Node field, remove it and click Apply.
   D. For each Oracle JDBC provider, repeat the following steps:

   1. Click the provider name.
   2. Select Data Sources.
   3. Click the name of the data source.
   4. Under the Oracle data source properties, modify the URL attribute according to Oracle RAC configuration:
   ```
   jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT =1521)) (ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICENAME)))
   ```
   5. Save your changes

8. Change to the directory `wp_profile_root/bin`.
9. Enter the following command to start the WebSphere Portal server:
   ```
   ./startServer.sh WebSphere_Portal
   ```

**Parent topic:** Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Assigning custom table spaces
Setting up a remote SQL Server database on Windows for a stand-alone production server

To setup a remote Microsoft SQL Server Enterprise Edition database, create user IDs and databases on a remote server. Tasks are provided to assist with creating the user IDs and the databases. Before you can use the tasks, you must modify properties files.

1. **Installing SQL Server**
   View the steps to install SQL Server for use with WebSphere Portal.

2. **Modifying database properties**
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

3. **Setting up databases**
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

   The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

5. **Configuring WebSphere Portal to use SQL Server 2005**
   View information on manually transferring data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

**Parent topic:** Configuring databases
Installing SQL Server

View the steps to install SQL Server for use with WebSphere Portal.

Before you begin this task, complete the following prerequisites:

- You should have completed reviewing the Planning for SQL Server topic.
- You must install SQL Server separately from WebSphere Portal.
- WebSphere Portal requires either the DataDirect Connect for JDBC 3.7 JDBC driver or the Microsoft SQL Server JDBC Driver.
- You can obtain the DataDirect Driver from DataDirect Technologies. The JDBC driver consists of two parts: the JDBC driver libraries and JTA libraries. The JDBC driver libraries must be put on the WebSphere Portal system, while the JTA libraries are intended for SQL Server.
- You can obtain the Microsoft SQL Server JDBC Driver from Microsoft. See the Microsoft SQL Server product documentation for installation details.

The following section provides instructions for installing SQL Server for use with IBM® WebSphere® Application Server and WebSphere Portal. These steps are the same for both the DataDirect and Microsoft drivers unless noted.

1. Install SQL Server and all required patches.
2. Select the authentication mode for this installation:
   - Windows Authentication Mode
   - Mixed Mode (Windows Authentication and SQL Server Authentication)
     Either authentication mode allows the user who is specified as the database administrator in the wkplc_comp.properties file to log in to the SQL server. Mixed Mode authentication allows either a Windows user or a SQL server user, or both, to log in to the SQL Server.
3. In the SQL Server Setup panel, Components to Install, select the following components, which are required services for WebSphere Portal:
   - SQL Server Database Services
     - Integration Services
       The option Integration Services, creates the directory DTS/Binn, where you will need to copy files when you install Microsoft SQL Server JDBC drivers and enable XA connections.
4. Complete the installation using SQL Server documentation as a guide.
5. Enable TCP/IP connectivity in the SQL Server Configuration Manager.
6. Install the JDBC driver using one of these methods:
   - Installing DataDirect Connect for JDBC drivers on UNIX
   - Installing DataDirect Connect for JDBC drivers on Windows
   - Installing Microsoft SQL Server JDBC drivers and enabling XA connections
7. Network the connections for MSDTC in the Windows Component Services as described in Microsoft Help and Support.
8. Start SQL Server.

**Installing DataDirect Connect for JDBC drivers on UNIX**

1. Purchase and download DataDirect Connect for JDBC and save file 360connectjdbc.jar in a temporary work directory.
To create the required files, run the following command from the directory that contains 360connectjdbc.jar:
```
jar -xvf 360connectjdbc.jar
```

2. Run `./Installer.sh` in the same directory.

3. When installation is complete, change directory to `/installation/lib`, where `installation` is the directory in which you installed the DataDirect Connect for JDBC drivers.

4. Run the following command to change the permissions on the JAR files in the `installation/lib` directory:
```
chmod 777 *.jar
```

5. Change the ownership and group of these files by running the following commands in the `installation/lib` directory:
```
chgrp system_grp *.jar
chown root *.jar
```

6. Change the ownership and group of these files by running the following commands in the `installation/lib` directory:
```
chgrp system_grp *.jar
chown root *.jar
```

Where `system_grp` is the system group as labeled by your operating system.

**Installing DataDirect Connect for JDBC drivers on Windows**

1. Purchase and download DataDirect Connect for JDBC and save file 360connectjdbc.jar in a temporary work directory.

2. To create the required files, run the following command from the directory that contains 360connectjdbc.jar:
```
jar -xvf 360connectjdbc.jar
```

3. Run `Installer.bat` in the same directory.

**Installing Microsoft SQL Server JDBC drivers and enabling XA connections**

1. Download and install the Microsoft SQL Server JDBC driver.

2. Copy file `sqljdbc_xa.dll` from the `xa` subdirectory to the `C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqljdbc_xa.dll` directory of the SQL Server installation.

3. Start the database server.

4. Ensure that the Distributed Transaction Coordinator has been started. The status can be verified in the list of services in the Computer Management console.

5. Start the Microsoft SQL Server Management Studio and connect to the local database engine as the system administrator, `sa`.

6. Select `File > Open > File` and select `xa_install.sql` from the subdirectory of the downloaded and extracted JDBC driver.

7. Execute the script by selecting `Query > Execute`. Note: Any warnings that appear in the messages section of the application window that say that stored procedures cannot be found can be safely ignored.

   A. Open the Windows Registry Editor (regedit) and navigate to the element `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSDTC\XADLL`
   B. From the menu bar, select `Edit > New > String Value` to create a new parameter named `sqljdbc_xa.dll` in that element.
   C. Change the value of the new parameter to the location of the `sqljdbc_xa.dll` file copied in Step 2 above, for example: `C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqljdbc_xa.dll`

**Enable XA Transactions in Windows Component Services**

1. From your Windows desktop, follow these steps:

2. Click `Start > Settings > Administrative Tools > Component Services`.
3. Expand the tree view to locate the computer where you want to turn on support for XA transactions (for example, My Computer).
4. Display the context menu for the computer name and click **Properties**.
5. Click **Options** and tune the Transaction Timeout that suits your environment. (The recommended minimum is 180 seconds).
6. Click **MSDTC** and click **Security Configuration**.
7. Under Security Settings, select **XA Transactions** to enable this support.
8. Click OK to save your changes.

**Note:** The installation documentation for JDBC XA connectivity refers to two known problems, see the Microsoft support site for more information:
- KB318818: Performance slows down when you use XA Transactions with SQL Server.

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server  
**Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Installing SQL Server

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:
  - `release.DbName=release`
  - `jcr.DbName=jcrdb`
  - `feedback.DbName=fdbkdb`
  - `likeminds.DbName=lmdb`
  - `community.DbName=comdb`
  - `customization.DbName=custdb`  
- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace `dbdomain` with:
  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
  - `dbdomain.DbType`
  - `dbdomain.DbName`
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.
   A. For dbdomainDbType, type sqlserver2005.
   B. For dbdomainDbName, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the dbdomainDbUrl property.
   C. For dbdomainDbSchema, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   D. For dbdomainDataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For dbdomainDbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:** The database element of this value should match the value of DbName.
   F. For dbdomainDbUser, type the user ID for the database administrator.
   G. For dbdomainDbPassword, type the password for the database administrator.
   H. For dbdomainDBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.
   I. For dbdomainDBA.DbPassword, type the database administrator password for privileged access operations during creation of the database.
   J. For dbdomainDbHome, type the root location for the database. **Note:**
      - This value is the location to store the database files locally.
      - This path must use two backslashes (\) instead of a forward slash (/).
   K. For dbdomainAdminUrl, type the sqlserver URL without a database attached. This value is used to connect to the server for database administration operations.
   L. For dbdomainDbHostName, type the hostname of the database.
3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important**: The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

   A. For `source.domain.DbType`, type of the database you are currently configured to use. The value for `source.domain.DbType` is Derby by default.
   
   B. For `source.domain.DbName`, type the name of the database domain you are currently using.
   
   C. For `source.domain.DbSchema`, type current schema identifier for objects within the database for this domain.
   
   D. For `source.domain.DataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.
   
   E. For `source.domain.DbUrl`, type the url currently used to access your database.
   
   F. For `source.domain.DbUser`, type the name of the user accessing this database.
   
   G. For `source.domain.DbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplcDbType.properties`.

   A. For `sqlserver2005.DbDriver`, type the name of the class that SqlProcessor uses to import SQL files.
   
   B. For `sqlserver2005.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.
   
   C. For `sqlserver2005.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.
   
   D. For `sqlserver2005.DbConnectionPoolDataSource`, type the name of the implementation class of the connection pool data source.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

---

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server  
**Previous topic:** Installing SQL Server  
**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

Before you begin, ensure that the following prerequisites are met:

- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

Prerequisites

- Installing SQL Server
- Modifying database properties

1. Change to the directory wp_profile_root\ConfigEngine
2. To create the databases, type the following command:
   ```
   ConfigEngine.bat create-database -DWasPassword=password
   ```
3. To create the database users, type the following command:
   ```
   ConfigEngine.bat setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires db_owner permission for the database user. The script that creates users is invoked by the configuration task setup-database and is located in the installation directory of WebSphere Portal:

   `<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql`

Creating users manually

Use this alternative method for creating users if you have problems running the setup-database task that is documented for setting up a remote SQL Server 2005 database on Windows for a stand-alone production server.

Parent topic: Setting up a remote SQL Server database on Windows for a stand-alone production server

Previous topic: Modifying database properties

Next topic: Linux stand-alone server: Assigning custom filegroups on SQL Server 2005
Creating users manually

Use this alternative method for creating users if you have problems running the setup-database task that is documented for setting up a remote SQL Server 2005 database on Windows for a stand-alone production server.

Before you begin:

- You should have completed installing SQL Server 2005 and creating databases.

- If WebSphere PortalVersion 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere PortalVersion 6.1.5 must be different than earlier versions to avoid conflicts during installation.

Create the recommended database users with the SQL Server Management Studio. At least one user is required for each SQL Server 2005 instance.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Connect to your SQL Server 2005 instance.
2. Expand the tree view beneath the SQL Server instance.
3. Expand Security and right-click on Logins.
4. In the opening context menu, select New Login....
5. Enter the database user names.
7. Set a password for the selected user.
8. In the Database Access window, select the database the user must connect to at runtime. The following mappings are recommended: Table 1. List of mappings by database

<table>
<thead>
<tr>
<th>Database</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELEASE</td>
<td>RELEASEUSR</td>
</tr>
<tr>
<td>COMMUNITY</td>
<td>COMMUNITYUSR</td>
</tr>
<tr>
<td>CUSTOMIZATION</td>
<td>CUSTOMIZATIONUSR</td>
</tr>
<tr>
<td>WORKFLOW</td>
<td>WORKFLOWUSR</td>
</tr>
<tr>
<td>WMM</td>
<td>WMMDBUSR</td>
</tr>
<tr>
<td>JCRDB</td>
<td>JCR</td>
</tr>
<tr>
<td>FDBKDB</td>
<td>FEEDBACK</td>
</tr>
<tr>
<td>LMDB</td>
<td>LIKEMINDS</td>
</tr>
</tbody>
</table>

9. Click OK to save the user changes.

10. Add the role SqlJDBCXAUser to all the database users that connect to the database using XA connections. This role is available for the database "master" only. To grant users this role:

A. Open the Microsoft SQL Server Management Studio and connect to the database instance.

B. Expand Security > Logins underneath the database instance name.

C. Click the user name to open the Login Properties window.

D. Select User Mapping.
E. Select database master in the database list.
F. Select SqlJDBCXAUser.
G. Click OK.

Parent topic: Setting up databases
Linux stand-alone server: Assigning custom filegroups on SQL Server 2005

The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

Before you begin:

- The custom filegroups must exist prior to the execution of the database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties` file.
- For details on creating filegroups refer to the documentation for the database.

**Prerequisites**

- Installing SQL Server
- Modifying database properties
- Setting up databases

If custom filegroups are assigned, each must be assigned explicitly. The default filegroups can be used to contain database objects; however the name of the default file group must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom filegroups:

1. Determine the names of your custom filegroups.
2. Open the mapping file `wp_profile_root/PortalServer/config/tablespaces` that specifies the table space and index space for each property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.indexspace.indexspace`
   
   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - release
     - community
     - customization
     - jcr
     - feedback
     - likeminds

3. Assign a filespace to each entry in the mapping file. The filegroup name must be prepended by the keyword `ON` and a space. For example: `community.COMP_INST.tablespace=ON COMM8KSPACE`
   
   Repeat this step for each domain that you are transferring.

4. Save and close `dbdomain.space_mapping.properties`.

5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - Windows: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
   - UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- /i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablesceMapping=true

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server

**Previous topic:** Setting up databases

**Next topic:** Configuring WebSphere Portal to use SQL Server 2005
Configuring WebSphere Portal to use SQL Server 2005

View information on manually transferring data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

**- Prerequisites**
- Installing SQL Server
- Modifying database properties
- Setting up databases

**Tips:**
- If you are transferring from Oracle, the *open_cursors* setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

**Steps for transferring data to another supported database**

1. Open a command prompt and change to the directory `wp_profile_root\ConfigEngine`
2. Enter the following commands to validate configuration properties. `ConfigEngine.bat validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password`
3. From the same command prompt as the previous steps, change to the directory `wp_profile_root\bin`
4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td>stopServer.bat server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td>stopServer.bat WebSphere_Portal -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

5. Transfer the database. **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.
A. Change to the directory `wp_profile_root\ConfigEngine`. 


B. Enter the following command: `ConfigEngine.bat database-transfer -DWasPassword=password`

**Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command: `ConfigEngine.bat database-transfer -DTransferDomainList=jcr -DWasPassword=password`

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. Change to the directory `wp_profile_root\bin`.

7. Enter the following command to start the WebSphere Portal server: `startServer.bat WebSphere_Portal`

8. Update the SQL Server 2005 statistics for Portal, and JCR databases by opening SQL Server Management Studio, selecting **New Query**, and running the following query: `use db_name exec sp_updatestats @resample='resample';`

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server

**Previous topic:** Linux stand-alone server: Assigning custom filegroups on SQL Server 2005
Verifying databases

After you configure IBM® WebSphere® Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.

**Prerequisites**

- **Technote for database connectivity issues**

You can verify the database connection using IBM WebSphere Application Server or by opening WebSphere Portal in a browser.

To verify that the WebSphere Portal application server is running by using WebSphere Application Server, complete these steps:

1. Open the WebSphere Application Server administrative console by entering the following address in a browser:
   
   \[http://hostname.example.com:10027/ibm/console\]
   
   where \(hostname.example.com\) is the fully qualified host name of the machine where WebSphere Portal is running and \(10027\) is the default transport port that is created by WebSphere Application Server.

2. Log into the administrative console.

3. Depending on your version of WebSphere Application Server, click the appropriate option:
   
   - For **WebSphere Application Server Version 6.1**: Click **Resources & JDBC Providers**.
   
   - For **WebSphere Application Server Version 7.0**: Click **Resources > JDBC > JDBC Providers**

4. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.

5. Select the name of the data source that is defined in \(wkplc_comp.properties\). The default data source is \(wpdbDS\).

6. Select the name of the JDBC provider that is specified in \(wkplc_dbtype.properties\). The default JDBC provider is \(wpdb JDBC_dbs\), where \(dbtype\) is replaced by the value that matches your environment.

7. Click **Test Connection** to verify the database connection. If configuration parameters have been changed, you might need to restart WebSphere Application Server for the test to complete.

   - To verify that the WebSphere Portal application server is running by opening WebSphere Portal in a browser, enter the following URL in a supported browser:
     
     \[http://hostname.example.com:10040/wps/portal\]
     
     where \(hostname.example.com\) is the fully qualified host name of the machine where WebSphere Portal is running and \(10040\) is the default transport port that is created by WebSphere Application Server.

**Parent topic:** Configuring databases
Preparing a remote Web server when portal is installed on Linux

Install and configure the Web server plug-in, provided by IBM® WebSphere® Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

- **Prerequisites**
  - Preparing your Linux operating system
  - Installing WebSphere Portal on Linux
  - Configuring databases

Perform the following steps to install and configure your Web server:

1. Install and configure the Web server; refer to the Web server documentation for information.
2. If using IBM Lotus® Domino®, edit the `NOTES.INI` file on the Web server. Set the `HTTPEnableConnectorHeaders` parameter to 1.
3. If using IBM HTTP Server or Apache Server, edit the `httpd.conf` file on the Web server. Set the `AllowEncodedSlashes` directive to `On`; the directive should be added at the root level as a global directive.
4. Stop the Web server.
5. Install and configure the Web server plug-in on the system where the Web server is located using the Plug-ins installation wizard provided with WebSphere Application Server. Refer to Selecting a Web server topology diagram and roadmap for information. **Note:** The WebSphere Application Server Web server instructions are for the default profile. If not using the default profile, refer to Configuring a Web server for a non-default profile.

If using WebDAV: After successfully installing the Web server plug-in, locate and open your `plugin-cfg.xml` file and set `AcceptAllContent` to `true`.

**Important:** Depending on how you use the Web server, you may need to adjust the `ServerIOTimeout` value, which defines how long the plug-in should wait for a response from the application. The recommended minimum value is 60 but you may need to adjust this value higher if you are retrieving data from a database. To update this value, locate and open your `plugin-cfg.xml` file and set `ServerIOTimeout` to a value that is appropriate for your business needs. For additional information, see Common questions about the Web server plug-in.

6. If you are using Sun Web Server Version 7 update 8, read Technote 1448262 and perform the steps to resolve the HTTP 408/409 error.
7. Start the Web server.

**Parent topic:** Setting up a stand-alone server on Linux

**Previous topic:** Configuring databases

**Next topic:** Configuring WebSphere Portal to use a user registry on Linux
Configuring WebSphere Portal to use a user registry on Linux

Configure user registry security on IBM® WebSphere® Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

Prior to configuring security, you should use the IBM WebSphere Application Server backupConfig task to create and store a backup of the IBM WebSphere Portal configuration; see backupConfig command for information.

- **Prerequisites**
  - Preparing your Linux operating system
  - Installing WebSphere Portal on Linux
  - Configuring databases

Perform the following tasks to configure WebSphere Portal to use a user registry:

1. **Preparing user registries on Linux**
   - Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

2. **Choosing your user registry model on Linux**
   - Choose between securing IBM WebSphere Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

3. **Adapting the attribute configuration**
   - After installing IBM WebSphere Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

4. **Stand-alone server: Configuring WebSphere Portal to use dynamic groups on Linux**
   - By default, WebSphere Portal is enabled for static groups. However, the Virtual Member Manager (VMM) allows users to be members of either static or dynamic groups. Static groups are those where a persistent binding exists between a group and its members. Dynamic groups are those where a search query is defined to retrieve the members of a group. If you have your LDAP server configured to use dynamic groups, complete the steps in this task for WebSphere Portal to use dynamic group queries when you setup your LDAP server.

5. **Stand-alone server: Enabling referrals for your LDAP user registry on Linux**
   - Referrals redirect object requests from one LDAP server to another when objects do not exist or cannot be located in a particular directory tree. You should enable referrals if your environment has more than one user registry existing on multiple servers or domains.

**Parent topic:** Setting up a stand-alone server on Linux

**Previous topic:** Preparing a remote Web server when portal is installed on Linux

**Next topic:** Tune your servers

**Related tasks**
- Managing your user registry on Linux
Preparing user registries on Linux

Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

Choose the appropriate LDAP server to install and setup:

- **Preparing a Domino Directory server**
  If you plan to use a Domino Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a SecureWay Security Server**
  If you plan to use a SecureWay Security Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Novell eDirectory**
  If you plan to use a Novell eDirectory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Sun Java System Directory Server**
  If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Tivoli Directory Server**
  If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

*Parent topic:* Configuring WebSphere Portal to use a user registry on Linux  
*Next topic:* Choosing your user registry model on Linux
Preparing a Domino Directory server

If you plan to use a Domino Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Domino Directory:

1. Perform the following steps to install the Domino Directory:
   A. Go to Lotus Domino documentation.
   B. Select the appropriate version tab for your product.
   C. Click the Download/View online link for the Lotus Domino for multiple platforms Information Center.
   D. Click Domino Administrator Help > Installation > Installing and setting up Domino servers > Server installation > Installing Domino on Linux and perform this task.
   E. Click Domino Administrator Help > Installation > Installing and setting up Domino servers > The Domino server setup program and perform this task.

2. Perform the following steps as a guide to create the WebSphere Portal administrative user:
   A. Navigate to the People view of the Domino Directory and then click Add Person.
   B. Enter the following values in the New Person form to create the wpsbind user:
      - Last Name
        - wpsbind
      - User Name
        - wpsbind/DominoDomain, where DominoDomain is your Lotus Domino Internet domain
        - wpsbind Note: Make sure you enter two values in the User Name field, where the first value includes the Lotus Domino domain.
      - Short name/UserID
        - wpsbind
      - Internet password
        - wpsbind
   C. Click Save and Close to save the new person record for wpsbind and return to the People view.
   D. Click Add Person and enter the following values in the New Person form to create the wpsadmin user:
      - Last Name
        - wpsadmin, where wpsadmin in the user ID for the WebSphere Portal Administrator
      - User Name
        - wpsadmin/DominoDomain, where DominoDomain is your Lotus Domino Internet domain
        - wpsadmin Note: Make sure you enter two values in the User Name field, where the first value includes the Lotus Domino domain.
      - Short name/UserID
        - wpsadmin
      - Internet password
        - wpsadmin
   E. Click Save and Close to save the new person record for wpsadmin and return to the People view.
   F. Navigate to the Groups view and click Add Group.
   G. Enter the following values in the New Group form on the Basic tab:
      - Group name
- **wpsadmins** Note: If you plan to configure WebSphere Portal for multiple user registries and your Lotus Domino LDAP will share a realm with another user registry, you must use the hierarchical naming convention for the group names, for example: wpsadmins/DominoDomain, to avoid unexpected results during WebSphere Portal runtime.

- **Group type**
  - Multi-purpose

- **Members**
  - wpsbind/DominoDomain
  - wpsadmin/DominoDomain

  Note: You can add additional administrator users if required.

H. Click **Save and Close** to save the wpsadmins group with the wpsbind and wpsadmin users as members.

3. Perform the following steps to update the access control list for the Domino Directory:
   A. Open the names.nsf file in the Lotus Domino Administrator or Lotus Notes client.
   B. Click **File > Application > Access Control** from the main menu to open the access control list for the file.
   C. In the **Access Control List > Basics** panel, ensure that the wpsadmins group has either Author or Editor access.
   D. Add the following Role Types to the wpsadmins group:
      - GroupCreator
      - GroupModifier
      - UserCreator
      - UserModifier
   E. Click **OK**.

**Parent topic:** Preparing user registries on Linux
Preparing a SecureWay Security Server

If you plan to use a SecureWay Security Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare SecureWay Security Server:

1. Install SecureWay Security Server; refer to the IBM SecureWay™ Security Server for z/OS® and OS/390® for information.

2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every dc=yourco,dc=com with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.
   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.

**Parent topic:** Preparing user registries on Linux
Preparing a Novell eDirectory

If you plan to use a Novell eDirectory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Novell eDirectory:

1. Install Novell eDirectory; refer to eDirectory for information.

2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the **Server Administration** folder, located in the left-hand navigation of the directory server console.
      2. Click the **Manage Server Properties** folder under the Server Administration folder and then select **Suffixes** on the main page.
      3. Type the **Base DN** name for the suffix; for example: `dc=yourcompany,dc=com`.
      4. Click **Add**.
      5. Click **OK** to save your changes.
      6. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the `PortalUsers.ldif` file as a working example and adapted appropriately to work with your LDAP server.
      - Use the `ContentUsers.ldif` file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every `dc=yourco,dc=com` with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the `objectclasses` to `accessGroup`. If using Tivoli Access Manager Version 6, set the `objectclasses` to `groupOfNames`.
   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on Linux
Preparing a Sun Java System Directory Server

If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Sun Java System Directory Server:

1. Install Sun Java System Directory Server; refer to Oracle software for information.

2. Perform the following steps to create the WebSphere Portal administrative user:
   
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: `dc=yourcompany,dc=com`.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.

   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   
   C. Replace every `dc=yourco,dc=com` with your suffix.
   
   D. Replace any prefixes and suffixes that are unique to your LDAP server.

   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.

   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the `objectclasses` to `accessGroup`. If using Tivoli Access Manager Version 6, set the `objectclasses` to `groupOfNames`.

   G. Save your changes.

   H. Follow the instructions provided with your directory server to import the LDIF file.

   I. Stop and restart the LDAP server.

**Parent topic:** Preparing user registries on Linux
Preparing a Tivoli Directory Server

If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Tivoli Directory Server:

   
   Note: Due to a restriction in Tivoli Directory Server, users or groups must not contain a Turkish uppercase dotted I or lowercase dotted i in the DN as this will prevent correct retrieval of that user or group.

2. Perform the following steps to create the WebSphere Portal administrative user:
   
   A. Optional: Perform the following steps to create a new directory suffix:
      
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.
   
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   
   C. Replace every dc=yourco,dc=com with your suffix.
   
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.
   
   G. Save your changes.
   
   H. Follow the instructions provided with your directory server to import the LDIF file.
   
   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on Linux
Choosing your user registry model on Linux

Choose between securing IBM® WebSphere® Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

Choose one of the following user registry models:

- **Configuring a stand-alone LDAP user registry on Linux**
  
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Configuring the default federated repository on Linux**
  
  Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

**Parent topic:** Configuring WebSphere Portal to use a user registry on Linux

**Previous topic:** Preparing user registries on Linux

**Next topic:** Adapting the attribute configuration
Configuring a stand-alone LDAP user registry on Linux

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Configuring a stand-alone LDAP user registry on Linux**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry to store all user account information for authorization.

- **Configuring a stand-alone LDAP user registry over SSL on Linux**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

**Parent topic:** Choosing your user registry model on Linux
Configuring a stand-alone LDAP user registry on Linux

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry to store all user account information for authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps. If you need to rerun the `wp-modify-ldap-security` task to change the LDAP repositories or because the task failed, you must choose a new name for the realm using the `standalone.ldap.realm` parameter or you can set `ignoreDuplicateIDs=true` in the `wkplc.properties` file, before rerunning the task.

Perform the following steps to configure a standalone LDAP user registry: **Note:** Use the `wp_security_XXX.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_security_XXX.properties` helper file.

1. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

2. Required: Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Stand-alone LDAP configuration heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.ldap.id`
   - `standalone.ldap.host`
   - `standalone.ldap.port`
   - `standalone.ldap.blindDN`
   - `standalone.ldap.blindPassword`
   - `standalone.ldap.ldapServerType`
   - `standalone.ldap.userIdMap`
   - `standalone.ldap.groupIdMap`
   - `standalone.ldap.groupMemberIdMap`
   - `standalone.ldap.userFilter`
   - `standalone.ldap.groupFilter`
   - `standalone.ldap.serverId`
   - `standalone.ldap.serverPassword`
   - `standalone.ldap.realm`
   - `standalone.ldap.primaryAdminId`
   - `standalone.ldap.primaryAdminPassword`
   - `standalone.ldap.primaryPortalAdminId`
   - `standalone.ldap.primaryPortalAdminPassword`
   - `standalone.ldap.primaryPortalAdminGroup`
   - `standalone.ldap.baseDN`
3. Required: Enter a value for the following required entity types parameters in the wkplc.properties file under the LDAP entity types heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - standaloneldap.et.group.objectClasses
   - standaloneldap.et.group.objectClassesForCreate
   - standaloneldap.et.group.searchBases
   - standaloneldap.et.personaccount.objectClasses
   - standaloneldap.et.personaccount.objectClassesForCreate
   - standaloneldap.et.personaccount.searchBases

4. Required: Enter a value for the following required group member parameters in the wkplc.properties file under the Group member attributes heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - standaloneldap.gm.groupMemberName
   - standaloneldap.gm.objectClass
   - standaloneldap.gm.scope
   - standaloneldap.gm.dummyMember

5. Required: Enter a value for the following required relative distinguished name (RDN) parameters in the wkplc.properties file under the Default parent, RDN attribute heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - standaloneldap.personAccountParent
   - standaloneldap.groupParent
   - standaloneldap.personAccountRdnProperties
   - standaloneldap.groupRdnProperties

6. Save your changes to the wkplc.properties file.

7. Optional: Enter the following Web content authors parameters in the wkplc_comp.properties file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. Note: See the wkplc_comp.properties file for specific information about the required parameters and for advanced parameters.
   - WcmContentAuthorsGroupId
   - WcmContentAuthorsGroupCN

8. Run the ./ConfigEngine.sh validate-standalone-ldap -DWasPassword=password task to validate your LDAP server settings. Attention: If you have not deleted the default file repository, WasPassword is the value entered during installation and not a value found in your LDAP user registry.
   Note: During the validation task, you may receive the following prompt: Add signer to the trust store now? Press y then Enter.

9. Run the ./ConfigEngine.sh wp-modify-ldap-security -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to set the stand-alone LDAP user registry.

10. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

11. Run the ./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to check that all defined attributes are available in the configured LDAP user registry. Important: When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

12. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the
13. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

**Parent topic:** Configuring a stand-alone LDAP user registry on Linux

**Related tasks**

- Adapting the attribute configuration
- Using the member fixer tool with IBM Lotus Web Content Management
- Starting and stopping servers, deployment managers, and node agents
Configuring a stand-alone LDAP user registry over SSL on Linux

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to configure a standalone LDAP user registry over SSL:

**Note:** Use the wp_security_xxx.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_security_xxx.properties helper file.

1. To specify the LDAP server's SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:

   **Server trust store**
   A. Add the certificate to the trust store:

   1. Log in to the WebSphere Application Server Administrative Console.
   2. Navigate to **Security > SSL certificate and key management > SSL configurations**.
   3. Click the appropriate SSL configuration from the list. For example, **NodeDefaultSSLSettings** for stand-alone environments or **CellDefaultSSLSettings** for clustered environments.
   4. Click **Key stores and certificates**.
   5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.
   6. Click **Signer certificates**, click **Add**, and then enter the following information:
      - Type the **Alias** the key store uses for the signer certificate.
      - Type the **File name** where the signer certificate is located.
   7. Click **OK** and then click **Save** to save the changes to the master configuration.

   **B. Retrieve the certificate from the port:**
   1. Log in to the WebSphere Application Server Administrative Console.
   2. Navigate to **Security > SSL certificate and key management > SSL configurations**.
   3. Click the appropriate SSL configuration from the list. For example, **NodeDefaultSSLSettings** for stand-alone environments.
Clustered environments: **CellDefaultSSLSettings**

Clustered environments: Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.

4. Click **Key stores and certificates**.
5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.
6. Click **Signer certificates**, click **Retrieve from port**, and then enter the following information:
   - Type the **Host** name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the SSL **Port** used when attempting to retrieve the signer certificate.
   - Type the **Alias** the key store uses for the signer certificate.
7. Click **Retrieve signer information** to retrieve the certificate from the port.
8. Click **OK** and then click **Save** to save the changes to the master configuration.

**Client trust storeNote:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

A. See Secure installation for client signer retrieval.

B. Run the **retrieveSigners** task from the **wp_profile_root/bin** directory; see **retrieveSigners command** for information. In a deployed environment, you will need to run the retrieveSigners task, for any federated node, against the Deployment Manager. **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

C. Update the trust store properties file:

1. Use a text editor to open the **ssl.client.props** file, located in the **wp_profile_root/properties** directory.
2. Change the **com.ibm.ssl.trustStore** parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter
   ```
   com.ibm.ssl.trustStore=${CONFIG_ROOT}\cells\wpsbvt\nodes\wpsbvt\trust.p12
   ```
   to use the default trust store.
3. Save your changes.

2. Use a text editor to open the **wkplc.properties** file, located in the **wp_profile_root/ConfigEngine/properties** directory.
3. Required: Enter a value for the following required parameters in the **wkplc.properties** file under the VMM Standalone LDAP configuration heading. **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - **standalone.idap.id**
   - **standalone.idap.host**
   - **standalone.idap.port**
   - **standalone.idap.blindDN**
   - **standalone.idap.bindPassword**
   - **standalone.idap.idapServerType**
   - **standalone.idap.userldMap**
   - **standalone.idap.groupIdMap**
   - **standalone.idap.groupMemberIdMap**
   - **standalone.idap.userFilter**
   - **standalone.idap.groupFilter**
   - **standalone.idap.serverId**
   - **standalone.idap.serverPassword**
   - **standalone.idap.realm**
   - **standalone.idap.principalAdminId**
   - **standalone.idap.principalAdminPassword**
   - **standalone.idap.primaryPortalAdminId**

761
4. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.ldap.et.group.objectClasses`
   - `standalone.ldap.et.group.objectClassesForCreate`
   - `standalone.ldap.et.group.searchBases`
   - `standalone.ldap.et.personaccount.objectClasses`
   - `standalone.ldap.et.personaccount.objectClassesForCreate`
   - `standalone.ldap.et.personaccount.searchBases`

5. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attributes heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.ldap.gm.groupMemberName`
   - `standalone.ldap.gm.objectClass`
   - `standalone.ldap.gm.scope`
   - `standalone.ldap.gm.dummyMember`

6. Required: Enter a value for the following required relative distinguished name (RDN®) parameters in the `wkplc.properties` file under the Default parent, RDN attribute heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.ldap.personAccountParent`
   - `standalone.ldap.groupParent`
   - `standalone.ldap.personAccountRdnProperties`
   - `standalone.ldap.groupRdnProperties`

7. Enter a value for the following parameters to enable Secure Socket Layers (SSL): **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - Required parameters:
     - `standalone.ldap.sslEnabled`
     - `standalone.ldap.sslConfiguration`
   - Optional parameters:
     - `standalone.ldap.certificateMapMode`
     - `standalone.ldap.certificateFilter`

8. Save your changes to the `wkplc.properties` file.

9. Run the `./ConfigEngine.sh validate-standalone-ldap -DWasPassword=password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, `WasPassword` is the value entered during installation and not a value found in your LDAP user registry. **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now? Press `y` then Enter.

10. Run the `./ConfigEngine.sh wp-modify-ldap-security -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to set the stand-alone LDAP user registry.

11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

12. Run the `./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured
LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

**Parent topic:** Configuring a stand-alone LDAP user registry on Linux

**Related tasks**
Adapting the attribute configuration
Starting and stopping servers, deployment managers, and node agents
Configuring the default federated repository on Linux

Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

Perform the following tasks to add user registries to the default federated repository:

- **Configuring a federated LDAP user registry on Linux**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Adding a database user registry on Linux**
  Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

- **Adding realm support on Linux**
  A realm is a group of users from one or more user registries that form a coherent group within IBM WebSphere Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

**Parent topic:** Choosing your user registry model on Linux
Configuring a federated LDAP user registry on Linux

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Adding an LDAP user registry on Linux**
  Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time. If IBM Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=groupName` and the hierarchical format is `cn=groupName,o=root`. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as `wpsadmin`, this ID cannot exist in the LDAP you are adding.

- **Adding an LDAP user registry over SSL on Linux**
  Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

**Parent topic:** Configuring the default federated repository on Linux
Adding an LDAP user registry on Linux

Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

If IBM® Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=groupName` and the hierarchical format is `cn=groupName,o=root`. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as `wpsadmin`, this ID cannot exist in the LDAP you are adding.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

1. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
2. Required: Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated LDAP Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - `federated.ldap.id`
   - `federated.ldap.host`
   - `federated.ldap.port`
   - `federated.ldap.bindDN`
   - `federated.ldap.bindPassword`
   - `federated.ldap.ldapServerType`
   - `federated.ldap.baseDN`

3. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - `federated.ldap.et.group.objectClasses`
   - `federated.ldap.et.group.objectClassesForCreate`
   - `federated.ldap.et.group.searchBases`
   - `federated.ldap.et.personaccount.objectClasses`
   - `federated.ldap.et.personaccount.objectClassesForCreate`
   - `federated.ldap.et.personaccount.searchBases`
4. Required: Enter a value for the following required group member parameters in the \texttt{wkplc.properties} file under the Group member attribute heading:\textbf{Note}: See the properties file for specific information about the required parameters and for advanced parameters.
   - \texttt{federated.ldap.gm.groupMemberName}
   - \texttt{federated.ldap.gm.objectClass}
   - \texttt{federated.ldap.gm.scope}
   - \texttt{federated.ldap.gm.dummyMember}

5. Save your changes to the \texttt{wkplc.properties} file.

6. Optional: Enter the following Web content authors parameters in the \texttt{wkplc_comp.properties} file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product.\textbf{Note}: See the \texttt{wkplc_comp.properties} file for specific information about the required parameters and for advanced parameters.
   - \texttt{WcmContentAuthorsGroupId}
   - \texttt{WcmContentAuthorsGroupCN}

7. Run the \texttt{./ConfigEngine.sh validate-federated-ldap -DWasPassword=\textit{password}} task to validate your LDAP server settings.\textbf{Attention}: If you have not deleted the default file repository, \texttt{WasPassword} is the value entered during installation and not a value found in your LDAP user registry.
   \textbf{Note}: During the validation task, you may receive the following prompt: \texttt{Add signer to the trust store now?}. Press \texttt{y} then \texttt{Enter}.

8. Run the \texttt{./ConfigEngine.sh wp-create-ldap -DWasPassword=\textit{password}} task, from the \texttt{wp_profile_root}/ConfigEngine directory, to add an LDAP user registry to the default federated repository.\textbf{Note}: Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: \textit{Starting and stopping servers, deployment managers, and node agents}.

10. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:
   A. Use a text editor to open the \texttt{wkplc.properties} file, located in the \texttt{wp_profile_root}/ConfigEngine/properties directory.
   B. Enter a value for the following required parameters in the \texttt{wkplc.properties} file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms:
      \textbf{Note}: See the properties file for specific information about the required parameters and for advanced parameters.
      - \texttt{id}
      - \texttt{baseDN}
      - \texttt{nameInRepository}
   C. Save your changes to the \texttt{wkplc.properties} file.
   D. Run the \texttt{./ConfigEngine.sh wp-create-base-entry -DWasPassword=\textit{password}} task, from the \texttt{wp_profile_root}/ConfigEngine directory, to create a base entry in a repository.
   E. Stop and restart all necessary servers to propagate your changes.

11. Optional: Run the \texttt{./ConfigEngine.sh wp-query-repository -DWasPassword=\textit{password}} task, from the \texttt{wp_profile_root}/ConfigEngine directory, to list the names and types of configured repositories.

12. Run the \texttt{./ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=\textit{password}} task, from the \texttt{wp_profile_root}/ConfigEngine directory, to check that all defined attributes are available in the configured LDAP user registry.\textbf{Important}: When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between
WebSphere Portal and the LDAP server.

13. Perform the following steps to update the user registry where new users and groups are stored:

   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - `personAccountParent`
      - `groupParent`
      - `personAccountRdnProperties`
      - `groupRdnProperties`

      The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:

      - `personAccountParent=dc=yourco,dc=com`
      - `groupParent=dc=yourco,dc=com`

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to delete the old attributes before adding the new attributes.

   E. Stop and restart all necessary servers to propagate your changes.

14. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for `realmName` or leave blank to update the default realm.

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

   E. Stop and restart all necessary servers to propagate your changes.

15. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

   **Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

   A. Edit the `wp_profile_root/PortalServer/wcm/shared/app/config/wcnservices/MemberFixerModule.properties` file.

   B. Add the following lines to the file:

   ```
   uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
   cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
   ```

   Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.
C. Save your changes and close the file.

D. Run the `./ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root`/ConfigEngine directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. **Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management**

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

16. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

17. Optional: **This step is required in a production environment.** Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

   **Important:**
   - Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
   - Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

   **Note:** If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

   A. Run the following task from the `wp_profile_root`/ConfigEngine directory: `./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword` **Important:** You must provide the full distinguished name (DN) for the `newAdminId` parameter.

   B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

   C. Run the following task from the `wp_profile_root`/ConfigEngine directory: `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `Dskip.ldap.validation=true` parameter to the task to skip the validation.

   D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

18. Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it. The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by...
duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: *Deleting the repository.*

**Parent topic:** Configuring a federated LDAP user registry on Linux

**Related tasks**
- Adapting the attribute configuration
- Deleting the repository on Linux
- Starting and stopping servers, deployment managers, and node agents

**Related information**
- User IDs and passwords
Adding an LDAP user registry over SSL on Linux

Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry over SSL to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

Note: Use the wp_add_federated_xxx.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_add_federated_xxx.properties helper file.

1. To specify the LDAP server's SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:
   - **Server trust store**
     A. Add the certificate to the trust store:
        1. Log in to the WebSphere Application Server Administrative Console.
        2. Navigate to Security > SSL certificate and key management > SSL configurations.
        3. Click the appropriate SSL configuration from the list. For example,
           - Stand-alone environments: NodeDefaultSSLSettings
           - Clustered environments: CellDefaultSSLSettings
           Clustered environments: Ensure the setting for SSL configuration for outbound connection matches your SSL settings.
        4. Click Key stores and certificates.
        5. Click the appropriate trust store from the list; for example, NodeDefaultTrustStore.
        6. Click Signer certificates, click Add, and then enter the following information:
           - Type the Alias the key store uses for the signer certificate.
           - Type the File name where the signer certificate is located.
        7. Click OK and then click Save to save the changes to the master configuration.
     B. Retrieve the certificate from the port:
        1. Log in to the WebSphere Application Server Administrative Console.
        2. Navigate to Security > SSL certificate and key management > SSL configurations.
        3. Click the appropriate SSL configuration from the list. For example,
           - Stand-alone environments: NodeDefaultSSLSettings
Clustered environments: 

Clustered environments: Ensure the setting for SSL configuration for outbound connection matches your SSL settings.

4. Click **Key stores and certificates**.

5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.

6. Click **Signer certificates**, click **Retrieve from port**, and then enter the following information:
   - Type the **Host** name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the **SSL Port** used when attempting to retrieve the signer certificate.
   - Type the **Alias** the key store uses for the signer certificate.

7. Click **Retrieve signer information** to retrieve the certificate from the port.

8. Click **OK** and then click **Save** to save the changes to the master configuration.

**Client trust store**

This task might report an error, but it does successfully update the trust store. You can ignore the error message.

A. See Secure installation for client signer retrieval.

B. Run the retrieveSigners task from the **wp_profile_root/bin** directory; see retrieveSigners command for information. In a deployed environment, you will need to run the retrieveSigners task, for any federated node, against the Deployment Manager. **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

C. Update the trust store properties file:

1. Use a text editor to open the **ssl.client.props** file, located in the **wp_profile_root/properties** directory.

2. Change the **com.ibm.ssl.trustStore** parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter
   ```
   com.ibm.ssl.trustStore=${CONFIG_ROOT}\cells\wpsbvt\nodes\wpsbvt\trust.p12
   ```
   to use the default trust store.

3. Save your changes.

2. Use a text editor to open the **wkplc.properties** file, located in the **wp_profile_root/ConfigEngine/properties** directory.

3. Required: Enter a value for the following required parameters in the **wkplc.properties** file under the VMM Federated LDAP Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - **federated.ldap.id**
   - **federated.ldap.host**
   - **federated.ldap.port**
   - **federated.ldap.bindDN**
   - **federated.ldap.bindPassword**
   - **federated.ldap.ldapServerType**
   - **federated.ldap.baseDN**

4. Required: Enter a value for the following required entity types parameters in the **wkplc.properties** file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - **federated.ldap.et.group.objectClasses**
   - **federated.ldap.et.group.objectClassesForCreate**
   - **federated.ldap.et.group.searchBases**
   - **federated.ldap.et.personaccount.objectClasses**
   - **federated.ldap.et.personaccount.objectClassesForCreate**
   - **federated.ldap.et.personaccount.searchBases**
Required: Enter a value for the following required group member parameters in the \texttt{wkplc.properties} file under the Group member attribute heading:

\begin{itemize}
    \item \texttt{federated.ldap.gm.groupMemberName}
    \item \texttt{federated.ldap.gm.objectClass}
    \item \texttt{federated.ldap.gm.scope}
    \item \texttt{federated.ldap.gm.dummyMember}
\end{itemize}

Note: See the properties file for specific information about the required parameters and for advanced parameters.

Enter a value for the following parameters to enable Secure Socket Layers (SSL):

Required parameters:

\begin{itemize}
    \item \texttt{federated.ldap.sslEnabled}
    \item \texttt{federated.ldap.sslConfiguration}
\end{itemize}

Optional parameters:

\begin{itemize}
    \item \texttt{federated.ldap.certificateMapMode}
    \item \texttt{federated.ldap.certificateFilter}
\end{itemize}

Save your changes to the \texttt{wkplc.properties} file.

Run the \texttt{./ConfigEngine.sh validate-federated-ldap -DWasPassword=password} task to validate your LDAP server settings. Attention: If you have not deleted the default file repository, \texttt{WasPassword} is the value entered during installation and not a value found in your LDAP user registry.

Note: During the validation task, you may receive the following prompt: \texttt{Add signer to the trust store now?}. Press \texttt{y} then \texttt{Enter}.

Run the \texttt{./ConfigEngine.sh wp-create-ldap -DWasPassword=password} task, from the \texttt{wp_profile_root}/ConfigEngine directory, to add an LDAP user registry to the default federated repository. Note: Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: \texttt{Starting and stopping servers, deployment managers, and node agents}.

Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:

A. Use a text editor to open the \texttt{wkplc.properties} file, located in the \texttt{wp_profile_root}/ConfigEngine/properties directory.

B. Enter a value for the following required parameters in the \texttt{wkplc.properties} file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms:

Note: See the properties file for specific information about the required parameters and for advanced parameters.

\begin{itemize}
    \item \texttt{id}
    \item \texttt{baseDN}
    \item \texttt{nameInRepository}
\end{itemize}

C. Save your changes to the \texttt{wkplc.properties} file.

D. Run the \texttt{./ConfigEngine.sh wp-create-base-entry -DWasPassword=password} task, from the \texttt{wp_profile_root}/ConfigEngine directory, to create a base entry in a repository.

E. Stop and restart all necessary servers to propagate your changes.

Optional: Run the \texttt{./ConfigEngine.sh wp-query-repository -DWasPassword=password} task, from the \texttt{wp_profile_root}/ConfigEngine directory, to list the names and types of configured repositories.
13. Run the `./ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

14. Perform the following steps to update the user registry where new users and groups are stored:

   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - `personAccountParent`
      - `groupParent`
      - `personAccountRdnProperties`
      - `groupRdnProperties`

      The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:

      - `personAccountParent=dc=yourco,dc=com`
      - `groupParent=dc=yourco,dc=com`

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to delete the old attributes before adding the new attributes.

   E. Stop and restart all necessary servers to propagate your changes.

15. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for `realmName` or leave blank to update the default realm.

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

   E. Stop and restart all necessary servers to propagate your changes.

16. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

    **Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

    A. Edit the `wp_profile_root /PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties` file.
B. Add the following lines to the file:
uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN

Replace portal_admin_DN with the distinguished name of the portal administrator and content_authors_group_DN with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the ./ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password task, located in the wp_profile_root/ConfigEngine directory. Note: Choose the appropriate value to enter for realm_name depending on the type of LDAP user registry you configured: Table 1. Value for realm_name when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for realm_name should match the value for standalone.ldap.realm in the wkplc.properties file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for realm_name should match the value for federated.realm in the wkplc.properties file. If the value for federated.realm is empty, use defaultWIMFileBasedRealm as the default value.</td>
</tr>
</tbody>
</table>

17. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

18. Optional: This step is required in a production environment. Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

   Important:
   - Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
   - Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

   Note: If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

   A. Run the following task from the wp_profile_root/ConfigEngine directory: ./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword Important: You must provide the full distinguished name (DN) for the newAdminid parameter.

   B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

   C. Run the following task from the wp_profile_root/ConfigEngine directory: ./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadminidgroupid Important: You must provide the full distinguished name (DN) for the newAdminid and newAdminGroupId parameters.

   Additional parameter for stopped servers: This task verifies the user against a running server instance. If the server is stopped, add the -Dskip.ldap.validation=true parameter to the task to skip the validation.

   D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

775
19. Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it. The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: *Deleting the repository.*

**Parent topic:** Configuring a federated LDAP user registry on Linux

**Related tasks**

Adapting the attribute configuration
Deleting the repository on Linux
Starting and stopping servers, deployment managers, and node agents

**Related information**

User IDs and passwords
Adding a database user registry on Linux

Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add a database user registry to the default federated repository; you must repeat these steps for each additional database user registry that you plan to add. **Note:** Use the `wp_add_DB.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wpcli.properties` file, you will use your `wp_add_DB.properties` helper file.

1. Set up a new database, including creating a new user with appropriate database privileges for accessing the database:

   **Instructions for setting up databases:** Refer to the appropriate documentation for the type of database you want to set up.

   **Consulting your database administrator:** The task of setting up a new database is typically performed by a database administrator. However, the following steps are provided for your reference in the event you are creating a stand-alone database for testing or demonstration purposes. Consult your database administrator before proceeding with the following steps if you plan to create a database for a production environment.

   **Table 1. Steps for creating a new database to use as a database user registry.**

<table>
<thead>
<tr>
<th>Database</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td>Perform the following steps to create a DB2 database: Install DB2. Enter the following database tuning commands: db2 &quot;CREATE DB dbname using codeset UTF-8 territory us PAGESIZE 8192&quot; db2 &quot;UPDATE DB CFG FOR dbname USING applheapsz 4096&quot; db2 &quot;UPDATE DB CFG FOR dbname USING app_ctl_heap_sz 1024&quot; db2 &quot;UPDATE DB CFG FOR dbname USING stmtimeap 32768&quot; db2 &quot;UPDATE DB CFG FOR dbname USING dbheap 2400&quot; db2 &quot;UPDATE DB CFG FOR dbname USING loglist 1000&quot; db2 &quot;UPDATE DB CFG FOR dbname USING logfilsiz 4000&quot; db2 &quot;UPDATE DB CFG FOR dbname USING logprimary 12&quot; db2 &quot;UPDATE DB CFG FOR dbname USING logsecond 20&quot; db2 &quot;UPDATE DB CFG FOR dbname USING logbufsz 32&quot; db2 &quot;UPDATE DB CFG FOR dbname USING avg_appls 5&quot; db2 &quot;UPDATE DB CFG FOR dbname USING locktimeout 30&quot; db2 &quot;UPDATE DB CFG FOR dbname using AUTO_MAINT off&quot;</td>
</tr>
</tbody>
</table>
Perform the following steps to define the `DbDriver` and `DbLibrary` parameter values:

A. Navigate to the following directory: `wp_profile_root/ConfigEngine/properties`
B. Locate and open `wkplc_dbtype.properties` with any text editor.
C. Enter a value for the following parameters under the appropriate database type properties heading:
   - `db_type.DbDriver`
   - `db_type.DbLibrary`
D. Save your changes.

3. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
4. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated Database Properties heading: **Note**: See the properties file for specific information about the required parameters and for advanced parameters.
   - `federated.db.DataSourceName`
   - `federated.dbDbType`
   - `federated.db.DbUrl`
   - `federated.db.Id`
   - `federated.db.baseDN`
   - `federated.db.DbUser`
   - `federated.db.DbPassword`
   - `federated.db.DbName`

5. Change the value for the `com.ibm.SOAP.requestTimeout` parameter to 1000.
   A. Navigate to the following directory: `wp_profile_root/properties`
   B. Locate and open `soap.client.props` with any text editor.
   C. Locate the `com.ibm.SOAP.requestTimeout` parameter and change the value to 1000.
   D. Save and close `soap.client.props`.

6. Perform the following steps in a clustered environment:
   A. Run the `.ConfigEngine.sh wp-prep-vmm-db-secured-environment -DWasPassword=password -DDbDomain=federated.db -Ddb_type.DmgrDbLibrary=localhost path of the database jars on the Deployment...`
Manager -DDmgrNodeName=dmgr_node_name task from the wp_profile_root/ConfigEngine directory to create the local Deployment Manager WebSphere variable used to access the database jars. **Note:** The db_type in db_type.DmgrDbLibrary should be set to the type of database you are using, for example db2. The local full path of the database jars on the Deployment Manager should be one of the following options:

- **DB2 Type 2 driver:** db2java.zip
- **DB2 Type 4 driver:** db2jcc.jar;db2jcc_license_cu.jar
- **DB2 for z/OS Type 2 driver:** db2java.zip
- **DB2 for z/OS Type 4 driver:** db2jcc.jar;db2jcc_license_cisuz.jar;db2jcc_javax.jar
- **Oracle:** ojdbc14.jar
- **SQL Server JDBC driver provided by Microsoft:** sqljdbc.jar
- **SQL Server JDBC driver provided by DataDirect:** sqlserver.jar;base.jar;util.jar

B. Run the following task. Include each node name as a comma separated list in the command:

**Running the task:** You do not have to run this task more than once. You can run this task from any node in the cluster.

1. Set the property value for federated.db.DbType in the wkplc.properties file if using a database user registry.
2. Run the ./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=password -DDbDomain=federated.db -DVmmNodeName=node_name -Db_type.NodeDbLibrary=local full path of the database jars task from the wp_profile_root/ConfigEngine directory on each node to create the variable used to access the VMM database jars. **Note:** VmmNodeName is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The db_type in db_type.NodeDbLibrary should be set to the type of database you are using, for example db2.

C. Stop and restart all necessary servers to propagate your changes.

7. Run the ./ConfigEngine.sh wp-create-db -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add a database user registry to the default federated repository. **Note:** Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

9. Perform the following steps to update the user registry where new users and groups are stored:

**Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored. 

**Attention:** During installation, the default file repository creates a default value in the personAccountRdnProperties and groupRdnProperties parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

B. Enter a value for the following required parameters in the wkplc.properties file under the VMM supported entity types configuration heading:

**Note:** See the properties file for specific information about the required parameters and for advanced parameters.
- personAccountParent
- groupParent
- personAccountRdnProperties
- groupRdnProperties

The parameters **groupParent** and **personAccountParent** must be set to the same value. For example:
- personAccountParent=dc=yourco,dc=com
- groupParent=dc=yourco,dc=com

C. Save your changes to the wkplc.properties file.

D. Run the `./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root`/ConfigEngine directory, to delete the old attributes before adding the new attributes.

E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Run the `./ConfigEngine.sh wp-query-repository -DWasPassword=password` task, from the `wp_profile_root`/ConfigEngine directory, to list the names and types of configured repositories.

**Parent topic:** Configuring the default federated repository on Linux

**Parent topic:** Updating your user registry on Linux
Adding realm support on Linux

A realm is a group of users from one or more user registries that form a coherent group within IBM® WebSphere® Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

Before configuring realm support, you must add all LDAP user registries and/or database user registries, that you will use to create a single realm or multiple realms, to the federated repository. If you are going to create multiple realms, you must create all required base entries within your LDAP user registries and/or database user registries. All base entry names must be unique within the federated repository. In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add realm support to your user registry model:

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
2. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading:
   - realmName
   - securityUse
   - delimiter
   - addBaseEntry

3. Save your changes to the wkplc.properties file.

4. Run the ./ConfigEngine.sh wp-create-realm -DWasPassword= password task, from the wp_profile_root/ConfigEngine directory, to add a new realm to the Virtual Member Manager configuration. Important: To create multiple realms, ensure that your federated repository contains the required unique base entries. Stop and restart the appropriate servers for your installation environment, and then update the wkplc.properties file with the base entry information and rerun the wp-create-realm task. Repeat these steps until all realms are created.

5. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

6. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading and then save your changes:
   - realmName
   - realm.personAccountParent
   - realm.groupParent
   - realm.orgContainerParent

7. Run the ./ConfigEngine.sh wp-modify-realm-defaultparents -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to update the default parents per entity type and realm. Important: Stop and restart the appropriate servers for your installation environment before rerunning this task for any additional entity types and realms.
8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents*.

9. Optional: Perform the following steps to add additional base entries to the realm configuration; for example, if you had two additional base entries (base entry 1 and base entry 2) to add to the realm you just created, you would update the wkplc.properties file with the information from base entry 1 and then run this task. Then you would update the properties file with the information for base entry 2 and then run this task:
   A. Use a text editor to open the wkplc.properties file, located in the `wp_profile_root/ConfigEngine/properties` directory.
   B. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading:
      - realmName
      - addBaseEntry
   C. Save your changes to the wkplc.properties file.
   D. Run the `./ConfigEngine.sh wp-add-realm-baseentry -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to add additional LDAP base entries to the realm configuration.
   E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID; this step is required if you change the default realm:
    A. Create a new user in the **Manage Users and Groups** portlet to replace the current WebSphere Application Server administrative user.
    B. Create a new user in the **Manage Users and Groups** portlet to replace the current WebSphere Portal administrative user.
    C. Create a new group in the **Manage Users and Groups** portlet to replace the current group.
    D. Run the `./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task, from the `wp_profile_root/ConfigEngine` directory, to replace the old WebSphere Application Server administrative user ID and group ID with the new user and group. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

E. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

F. Run the `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task to replace the old WebSphere Portal administrative user ID and group ID with the new user and group. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

G. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

11. Optional: Perform the following steps to set the realm you created as the default realm. **Remember:** Only users defined in base entries that exist in the default realm are able to log into WebSphere Portal. If you find that a user cannot log in to WebSphere Portal, check to see if the base entry that contains the user exists in the default realm. You can run the `wp-query-realm-baseentry` task to see what base entries are part of the default realm. If the default realm is missing the base entry, run the `wp-add-realm-baseentry` task to add the base entry to the default realm.
A. Use a text editor to open the `wkplc.properties` file, located in the `<wp_profile_root>/ConfigEngine/properties` directory.

B. For `defaultRealmName`, type the `realmName` property value you want to use as the default realm.

C. Save your changes to the `wkplc.properties` file.

D. Run the `./ConfigEngine.sh wp-default-realm -DWasPassword=password` task, from the `<wp_profile_root>/ConfigEngine` directory, to set this realm as the default realm.

E. Stop and restart all necessary servers to propagate your changes.

12. Optional: Perform the following steps to query a realm for a list of its base entries:

   A. Use a text editor to open the `wkplc.properties` file, located in the `<wp_profile_root>/ConfigEngine/properties` directory.

   B. For `realmName`, type the name of the realm you want to query.

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-query-realm-baseentry -DWasPassword=password` task, from the `<wp_profile_root>/ConfigEngine` directory, to list the base entries for a specific realm.

13. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

   A. Use a text editor to open the `wkplc.properties` file, located in the `<wp_profile_root>/ConfigEngine/properties` directory.

   B. Enter a value for `realmName` or leave blank to update the default realm.

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `<wp_profile_root>/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

   E. Stop and restart all necessary servers to propagate your changes.

**Parent topic:** Configuring the default federated repository on Linux
Adapting the attribute configuration on Linux

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

After installation, IBM WebSphere Portal has a predefined set of attributes for users and groups. Your LDAP server may have a different set of predefined user and group attributes. To ensure proper communication between WebSphere Portal and your LDAP server, you can configure additional attributes and flag existing attributes as required or unsupported on a per repository basis or for all configure repositories.

LDAP servers can only handle attributes that are explicitly defined in their schema. The LDAP server’s schema is different from the WebSphere Portal schema but the two schemas should match for proper communication between WebSphere Portal and the LDAP server. The task to add the LDAP user registry does some basic attribute configurations depending on the type of LDAP server that you choose. You may, however, still need to adapt the WebSphere Portal configuration to match the LDAP schema; for example, if an attribute is defined in WebSphere Portal but not in the LDAP server, you will need to perform one of the following tasks to resolve this mismatch:

- Flag the attribute as unsupported for the LDAP server
- Introduce an attribute mapping that maps the WebSphere Portal attribute to an attribute defined in the LDAP schema

Perform the following tasks to adapt the attribute configuration to match the configured LDAP server(s) and your business needs:

1. **Querying the defined attributes on Linux**
   After installing IBM WebSphere Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

2. **Adding attributes on Linux**
   The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM WebSphere Portal and your user registry.

3. **Mapping attributes**
   After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

4. **Removing attributes**
   Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

Parent topic: Configuring WebSphere Portal to use a user registry on Linux
Previous topic: Choosing your user registry model on Linux
Next topic: Stand-alone server: Configuring WebSphere Portal to use dynamic groups on Linux

Related tasks
Adding an LDAP user registry on Linux
Adding an LDAP user registry over SSL on Linux
Configuring a stand-alone LDAP user registry on Linux
Configuring a stand-alone LDAP user registry over SSL on Linux
Querying the defined attributes on Linux

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

Warning: If you are using a database user registry or a property extension database, copy the database drivers to the AppServer_root/lib directory before executing this script.

Run the ./ConfigEngine.sh wp-query-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, any time during the configuration process or at runtime to query an overview of the currently defined attributes. This task creates the availableAttributes.html report, located in the wp_profile_root/ConfigEngine/log directory. The report contains one table that lists the available attributes for Users (PersonAccount) and one table that lists the available attributes for Groups. For each configured repository there is a column that indicates if the attribute is flagged as unsupported or if the attribute is mapped to a different LDAP attribute. Note: This task does not validate the existence of attributes in the LDAP schema.

Parent topic: Adapting the attribute configuration

Next topic: Adding attributes on Linux
Adding attributes on Linux

The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM® WebSphere® Portal and your user registry.

Perform the following steps to add new attributes to your user registry:

1. Install the required Enterprise Archive (.ear) file on WebSphere Application Server. Select the appropriate task to install the .ear file:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone environment</td>
<td>Perform the following steps to install the .ear file in a stand-alone environment: Open a command prompt. Navigate to the wp_profile_root/ConfigEngine directory. Run the ./ConfigEngine.sh wp-la-install-ear -DWasPassword=password task.</td>
</tr>
<tr>
<td>Clustered environment</td>
<td>Perform the following steps to install the .ear file in a clustered environment: Open a command prompt. Navigate to the wp_profile_root/ConfigEngine directory on the primary node. Run the ./ConfigEngine.sh wp-la-install-ear -DWasPassword=dmgr_password -DServerName=dmgr -DNodeName=node_name task. Where node_name is the name of the node where the deployment manager resides; you can find the node_name value in the WebSphere Application Server Administrative Console under System administrator &gt; Deployment Manager &gt; Runtime tab &gt; General Properties &gt; Node Name.</td>
</tr>
</tbody>
</table>

2. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

3. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

4. Enter a value for the following required parameters in the wkplc.properties file under the VMM Property Extension Properties heading:

   Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - la.providerURL
   - la.propertyName
   - la.entityTypes
   - la.dataType
   - la.multiValued

5. Save your changes to the wkplc.properties file.

6. Run the ./ConfigEngine.sh wp-add-property -DWasPassword=password task to add the attribute to the user registry.

   Note: This task performs an EJB call to WebSphere Application Server, which must authenticate against WebSphere...
Application Server. Depending on the configuration in the sas.client.props file, you may receive a popup window or a command line prompt asking for user identity and password. Enter the WebSphere Application Server user ID and password. **Remember:** If you have multiple properties to add, repeat all steps, except for the wp-la-install-ear task, until all new attributes are added.

7. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents.*

**Parent topic:** Adapting the attribute configuration  
**Previous topic:** Querying the defined attributes on Linux  
**Next topic:** Mapping attributes

**Related tasks**
Starting and stopping servers, deployment managers, and node agents
Mapping attributes

After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

Perform the following steps to map attributes between WebSphere Portal and your LDAP server; if you have multiple LDAP servers, you will need to perform these steps for each LDAP server:

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
2. Enter a value for one of the following sets of parameters in the wkplc.properties file to identify your LDAP server:
   - Note: Make sure you use the same values you used to configure your LDAP server.

Table 1. Identifying your LDAP server in the wkplc.properties file.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>The following parameters are found under the LDAP attribute configuration heading. Note: See the properties file for specific information about the required parameters and for advanced parameters. standalone.ldap.id standalone.ldap.host standalone.ldap.port standalone.ldap.sslEnabled standalone.ldap.bindDN standalone.ldap.bindPassword standalone.ldap.baseDN</td>
</tr>
<tr>
<td>Federated</td>
<td>The following parameters are found under the VMM Federated repository properties heading. Note: See the properties file for specific information about the required parameters and for advanced parameters. federated.ldap.id federated.ldap.host federated.ldap.port federated.ldap.sslEnabled federated.ldap.bindDN federated.ldap.bindPassword federated.ldap.baseDN</td>
</tr>
</tbody>
</table>

3. Run one of the following tasks to check that all defined attributes are available in the configured LDAP user registry:

Table 2. Task to check that all defined attributes are available in the configured LDAP user registry.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory.</td>
</tr>
<tr>
<td>Federated</td>
<td>./ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory.</td>
</tr>
</tbody>
</table>

4. Open the ConfigTrace.log file, located in the wp_profile_root/ConfigEngine/log directory, to review the following output for the PersonAccount and Group entity type:
- The following attributes are defined in WebSphere Portal but not in the LDAP server
  - This list contains all attributes that are defined in WebSphere Portal but not available in the LDAP. Flag attributes that you do not plan to use in WebSphere Portal as unsupported. Map the attributes that you plan to use to the attributes that exist in the LDAP; you must also map the `uid`, `cn`, `firstName`, `sn`, `preferredLanguage`, and `ibm-primaryEmail` attributes if they are contained in the list.

- The following attributes are flagged as required in the LDAP server but not in WebSphere Portal
  - This list contains all attributes that are defined as "MUST" in the LDAP server but not as required in WebSphere Portal. You should flag these attributes as required within WebSphere Portal; see the step below about flagging an attribute as either unsupported or required.

- The following attributes have a different type in WebSphere Portal and in the LDAP server
  - This list contains all attributes that WebSphere Portal might ignore because the data type within WebSphere Portal and within the LDAP server do not match.

5. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

6. Enter a value for one of the following sets of parameters in the `wkplc.properties` file to correct any issues found in the config trace file: **Table 3. Parameters to define in the wkplc.properties file to correct any issues found in the config trace file.**

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Parameters</th>
</tr>
</thead>
</table>
| Stand-alone     | The following parameters are found under the LDAP attribute configuration headiing:**Note:** See the properties file for specific information about the required parameters and for advanced parameters.  
`standalone.idldap.idstandalone.ldap.attributes.nonSupportedstandalone.ldap. attributes.nonSupported.deletestandalone.ldap.attributes.mapping.ldapNamestandalone.ldap.attributes.mapping.portalNamestandalone.ldap.attributes.mapping.entityTypes`For example, the following values will flag certificate and members as unsupported attributes and will map `ibm-primaryEmail` to `mail` and `ibm-jobTitle` to `title` for both the PersonAccount and Group entityTypes.  
`standalone.idldap.attributes.nonSupported=certificate, members`  
`standalone.idldap.attributes.mapping.ldapName=mail, title`  
`standalone.idldap.attributes.mapping.portalName=ibm-primaryEmail, ibm-jobTitle`  
`standalone.idldap.attributes.mapping.entityTypes=PersonAccount, Group`
7. Save your changes to the wkplc.properties file.

8. Run one of the following tasks to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between WebSphere Portal and the LDAP user registry:

   **Table 4. Task to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between Portal and the LDAP user registry.**

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>./ConfigEngine.sh wp-update-standalone-ldap-attribute-config -DWasPassword=\password\ task, from the \wp_profile_root\ConfigEngine directory</td>
</tr>
<tr>
<td>Federated</td>
<td>./ConfigEngine.sh wp-update-federated-ldap-attribute-config -DWasPassword=\password\ task, from the \wp_profile_root\ConfigEngine directory</td>
</tr>
</tbody>
</table>

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

10. Optional: Perform the following steps to flag an attribute as either unsupported or required for the entire WebSphere Portal environment instead of just for the specified LDAP:

   A. Enter a value for the following required parameters in the wkplc.properties file:

   - `user.attributes.required`
   - `user.attributes.nonsupported`

   Note: See the properties file for specific information about the required parameters and for advanced parameters.

   B. Save your changes to the wkplc.properties file.

   C. Run the `./ConfigEngine.sh wp-update-attribute-config -DWasPassword=\password\ task`, from the \wp_profile_root\ConfigEngine directory.

   D. Stop and restart all necessary servers to propagate your changes.

**Parent topic:** Adapting the attribute configuration

**Previous topic:** Adding attributes on Linux
Related tasks
Starting and stopping servers, deployment managers, and node agents
Removing attributes

Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

Perform the following steps to remove an attribute from your database:

**Important:** Do not remove attributes that have already been populated with user values because this can cause database inconsistencies.

**Cluster note:** In a clustered environment, perform the following steps on the deployment manager and then resynch the nodes.

1. Perform the following steps to remove an attribute stored in a property extension database; this is an attribute that was added using the `wp-add-la-property` task.
   - A. Open the tool you use to edit your database.
   - B. Verify that your attribute name is available in the LAPROP table.
   - C. Delete the required attributes from the LAPROP table.
   - D. Open the `wimxmlextension.xml` file, located in the `wp_profile_root/config/cells/cellname/wim/model` directory.
   - E. Locate and delete the `propertySchema` definition for the attributes that you deleted from the LAPROP table; for example:
     ```xml
     <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String" multiValued="true" propertyName="attribute_name">
     <wim:applicableEntityTypeName>PersonAccount</wim:applicableEntityTypeName>
     </wim:propertySchema>
     ```
   - F. Save your changes to the `wimxmlextension.xml` file.
   - G. Open the `wimconfig.xml` file, located in the `wp_profile_root/config/cells/cellname/wim/config` directory.
   - H. Locate and delete the `propertiesNotSupported` definitions for the attributes that you deleted from the LAPROP table; for example:
     ```xml
     <config:propertiesNotSupported name="attribute_name"/>
     ```
   - I. Save your changes to the `wimconfig.xml` file.
   - J. Stop and restart the server1 and WebSphere_Portal servers from the `wp_profile_root/bin` directory.

2. Perform the following steps to remove an attribute that is not stored in a property extension database:
   - A. Open the `wimxmlextension.xml` file.
   - B. Locate and delete the `propertySchema` definition for the attributes you previously added; for example:
     ```xml
     <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String" multiValued="true" propertyName="attribute_name">
     <wim:applicableEntityTypeName>PersonAccount</wim:applicableEntityTypeName>
     </wim:propertySchema>
     ```
   - C. Save your changes to the `wimxmlextension.xml` file.
   - D. Open the `wimconfig.xml` file.
   - E. Locate and delete the stanza that corresponds to the custom attribute you deleted from the `wimextension.xml` file; for example:
     ```xml
     <config:attributes name="attribute_name" propertyName="property_name">
     <config:entityTypes>PersonAccount</config:entityTypes>
     </config:attributes>
     ```
   - F. Save your changes to the `wimconfig.xml` file.
G. Stop and restart the server1 and WebSphere_Portal servers.

Parent topic: Adapting the attribute configuration
Previous topic: Mapping attributes
Stand-alone server: Configuring WebSphere Portal to use dynamic groups on Linux

By default, WebSphere Portal is enabled for static groups. However, the Virtual Member Manager (VMM) allows users to be members of either static or dynamic groups. Static groups are those where a persistent binding exists between a group and its members. Dynamic groups are those where a search query is defined to retrieve the members of a group. If you have your LDAP server configured to use dynamic groups, complete the steps in this task for WebSphere Portal to use dynamic group queries when you setup your LDAP server.

Perform the required tasks to configure either a stand-alone or federated LDAP server security.

The steps in this task use `groupOfURLs` as the object class for dynamic groups and `memberURL` as the dynamic membership attribute. The actual values for object classes and dynamic membership attributes can vary depending on your LDAP server. For this reason, you should export an LDIF file to verify the object classes and dynamic membership attributes. Either refer to your LDAP documentation or ask your LDAP administrator for instructions on exporting an LDIF file.

**Clustered environments:** Perform the following steps on the Deployment Manager then synchronize the nodes.

To configure WebSphere Portal to use dynamic groups, do the following:

1. Choose the appropriate set of steps, depending on your LDAP server environment: *Table 1. Steps for enabling dynamic groups*

<table>
<thead>
<tr>
<th>LDAP server environment</th>
<th>Steps to perform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone LDAP server or federated LDAP server(s)</td>
<td>Navigate to the following directory: \wp_profile_root\cells\cell_name\wim\config. Locate and open wimconfig.xml with any text editor. Add the following line to the <code>&lt;config:groupConfiguration&gt;</code> tag: <code>&lt;config:dynamicMemberAttributes name=&quot;memberurl&quot; objectClass=&quot;groupofurls&quot;/&gt;</code>. Save and close wimconfig.xml.</td>
</tr>
<tr>
<td>Federated LDAP server(s)</td>
<td>Log in to the administration console. Select <strong>Security &gt; Secure administration, applications, and infrastructure.</strong> Under <strong>Available realm definitions</strong>, select <strong>Federated repositories</strong> and click <strong>Configure</strong>. Under <strong>Related items</strong>, click <strong>Manage repositories</strong>. Select the appropriate repository from the list. Under <strong>Additional Properties</strong>, click <strong>Group attribute definition</strong> then click <strong>Dynamic member attributes</strong>. Click <strong>New</strong> and specify values for the <strong>Name</strong> and <strong>Object class</strong> fields as appropriate. For example, <strong>Name</strong>: <code>memberurl</code> <strong>Object class</strong>: <code>groupofurls</code> Click <strong>OK</strong> and save the changes to the master configuration.</td>
</tr>
</tbody>
</table>

2. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under **Related tasks**: *Starting and stopping servers, deployment managers, and node agents.*
Stand-alone server: Enabling referrals for your LDAP user registry on Linux

Referrals redirect object requests from one LDAP server to another when objects do not exist or cannot be located in a particular directory tree. You should enable referrals if your environment has more than one user registry existing on multiple servers or domains.

To configure your portal to use LDAP referrals, do the following:

1. Use any text editor to open the wkplc.properties file in the following directory: wp_profile_root/ConfigEngine/properties.
2. Specify values for the following parameters:
   - `et.ldap.id=ID_of_your_LDAP_server`
   - `et.ldap.host=hostname_of_your_LDAP_server`
   - `et.ldap.referral=follow`
3. Save and close wkplc.properties.
4. Run the following task from the wp_profile_root/ConfigEngine directory to create an LDAP entity type:
   - UNIX: `./ConfigEngine.sh wp-update-et-ldap -DWasPassword=password`
   - Windows: `ConfigEngine.bat wp-update-et-ldap -DWasPassword=password`
   - IBM® i5/OS: `ConfigEngine.sh wp-update-et-ldap -DWasPassword=password`
5. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

Parent topic: Configuring WebSphere Portal to use a user registry on Linux
Previous topic: Stand-alone server: Configuring WebSphere Portal to use dynamic groups on Linux
Tune your servers

Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

- Prerequisites
  - Preparing your Linux operating system
  - Installing WebSphere Portal on Linux
  - Configuring databases
  - Base Portal Tuning Scenarios

Tuning a WebSphere Portal environment involves tuning and configuring the various systems and components of the environment. The tuning guide provides general concepts and details specifics configuration instructions. Instructions are included for:

- Configuring the application server and the resources defined for that application server
- Determining the cloning strategy for expanding or extending the environment
- Tuning the database(s) and database server
- Tuning the directory server and its database
- Tuning the Web server
- Tuning the operating system and network
- Tuning the WebSphere Portal services

Parent topic: Setting up a stand-alone server on Linux

Previous topic: Configuring WebSphere Portal to use a user registry on Linux

Related information
- WebSphere Portal and Lotus Web Content Management 6.1.x Performance Tuning Guide
- IBM WebSphere Portal Performance Troubleshooting Guide
Setting up a stand-alone server on Solaris

After you have prepared the operating system you are ready to install WebSphere Portal.

- **Prerequisites**

  - [Technotes for installation and configuration issues](#)

  Select the operating system on which you are installing WebSphere Portal.

1. **Preparing your Solaris operating system**

   View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

2. **Installing WebSphere Portal on Solaris**

   IBM WebSphere Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

3. **Configuring databases**

   Before transferring default data to the production database server, you need to prepare the database server. Database preparation includes creating user IDs and databases. For this installation path, stand-alone production server, remote databases servers are used.

4. **Preparing a remote Web server when portal is installed on Solaris**

   Install and configure the Web server plug-in, provided by IBM WebSphere Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

5. **Configuring WebSphere Portal to use a user registry on Solaris**

   Configure user registry security on IBM WebSphere Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

6. **Tune your servers**

   Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

**Parent topic:** Setting up a stand-alone production server
Preparing your Solaris operating system

View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

Perform the following steps to prepare your operating system:

1. Perform the following steps to prepare your Solaris kernel settings: Several Solaris kernel values are typically too small for the messaging requirements of WebSphere Portal. Starting the internal JMS server or client with insufficient kernel resources produces a First Failure Support Technology (FFST) file in the `/var/errors` directory. Before installing WebSphere Portal, review the machine's configuration.

   **Note:** The values described here are a starting point for messaging in WebSphere Portal only. If your system has other applications installed, the value requirements will likely be different. For example, if values that are already set are higher than the settings listed here, the values should not be lowered. Be sure to check the requirements made on `/etc/system` by other already-installed applications before altering existing values.

   A. Type the `sysdef -i` command to review the configuration.
   B. Set `shmsys:shminfo_shmmax` (valid for Solaris Version 9 only) to 4294967295.
   C. Set `shmsys:shminfo_shmmni` (valid for Solaris Version 9 only) to 1024.
   D. Set `semsys:seminfo_semaem` (valid for Solaris Version 9 only) to 16384.
   E. Set `semsys:seminfo_semmni` (valid for Solaris Version 9 only) to 1024.
   F. Set `semsys:seminfo_semmns` (valid for Solaris Version 9 only) to 16384.
   G. Set `semsys:seminfo_semmni` (valid for Solaris Version 9 only) to 100.
   H. Set `semsys:seminfo_semopm` (valid for Solaris Version 9 only) to 100.
   I. Set `semsys:seminfo_semnu` (valid for Solaris Version 9 only) to 2048.
   J. Set `msgsys:msginfo_msgmax` (valid for Solaris Version 9 only) to 65535.
   K. Set `rlim_fd_cur` to 1024.
   L. Set `rlim_fd_max` to 10240.
   N. Reboot the operating system to apply the updates.

2. **Web Content Management only:** Use the `ulimit -f` command to set the maximum size of files that can be created to be at least the size of the largest file you would need to upload to the content server. The command `ulimit -f -1` removes any limit on file size.

3. Set the file descriptor limit to 10240; for example, `ulimit -n 10240`.

4. Perform the following steps to prepare for non-global zone:

   A. Do not inherit package directories when you create the non-global zone because the inherited software packages will be read-only.
   B. Stop WebSphere Portal and all related processes before installing or uninstalling.
   C. Verify that the following processes are stopped:
      - `/opt/IBM/WebSphere/AppServer/java/bin/java`
      - `/opt/IBM/WebSphere/AppServer/java/jre/bin/java`

**Parent topic:** Setting up a stand-alone server on Solaris

**Next topic:** Installing WebSphere Portal on Solaris
Installing WebSphere Portal on Solaris

IBM® WebSphere® Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

Review the Supported hardware and software requirements for your operating system and ensure that all hardware and software matches the minimum product level before installing WebSphere Portal. Review Installation methods, options, and sources before installing to determine your installation source location and installation method. Note: If you plan to use IBM WebSphere Application Server Version 7.0, you must purchase and install it before installing WebSphere Portal on the existing WebSphere Application Server Version 7.0 instance.

Perform the following steps to install WebSphere Portal:

1. Type `ping yourserver.yourcompany.com` on a command line to verify that your fully qualified host name is properly configured.
2. Type `ping localhost` on a command line to verify that your network is properly configured.
3. If you are installing with an existing WebSphere Application Server instance, ensure it is installed at the supported level and has the following features installed:
   - Application Server
   - Administration
   - Scripted Administration
   - Administrative Console
   - Ant and Deployment Tools
   - Deploy Tool
   - Ant Utilities
   Restriction: Installing WebSphere Portal Version 6.1.0.1 or higher on an existing WebSphere Application Server Version 7.0 instance where WebSphere Portal Version 7.0 was installed is unsupported.

Important: Besides ensuring that the existing WebSphere Application Server instance is at the supported level, you will also need to ensure that Apache Derby is installed at the supported level before installing WebSphere Portal. See WebSphere Portal detailed system requirements for supported levels.

- Application Server
- Administration
- Scripted Administration
- Administrative Console
- Ant and Deployment Tools
- Deploy Tool
- Ant Utilities

4. Optional: Perform the following steps if you want to install WebSphere Portal using a non-root user:
   Restriction: Although it is possible to install WebSphere Application Server as a non-root user, there are limitations to this option that can affect WebSphere Portal functionality, which is why you should install WebSphere Application Server as a root user.
   - Install the current supported version of WebSphere Application Server as a root user.
   - Open a command prompt and perform the following steps to create a non-root user and to change ownership:
     1. Use the appropriate system commands to create a new group, to create a new user, and to add the new user to the new group.
     2. Run the following tasks to change the rights of the non-root user:

        ```
        chgrp -R group_name /opt/IBM
        chmod -R g+wr /tmp
        ```

chgrp -R group_name /tmp
chmod -R g+wr /var/tmp
chgrp -R group_name /var/tmp

C. Launch the installation program per the next step. During the installation, select the existing WebSphere Application Server using the non-root user and set the installation location to a unique location under the path where non-root permissions were granted.

5. Choose one of the following installation commands based on the installation method you want to use to install WebSphere Portal; see Installation Methods for more information:

- **Restriction:** When defining your installation path, the ONLY supported characters are ASCII letters [A-Z and a-z], numbers [0-9], slashes [ / or \
, depending on your operating system], and the dash [-].

- **Advance installation parameters:** To customize your installation, you can add various parameters to your installation commands; for example, you can specify your port information. See Advanced installation parameters for information.

### Table 1. Installation task options

<table>
<thead>
<tr>
<th>Installation type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphical user interface</td>
<td>./install.sh</td>
</tr>
<tr>
<td>Console mode</td>
<td>./install.sh -console from the PortalExpress subdirectory</td>
</tr>
<tr>
<td>Silent install</td>
<td>./install.sh -options &quot;path_to_file/response_filename&quot;, where path_to_file is the full path to the response file and response_filename is the name of the file. A sample install response file (installresponse.txt) and a sample uninstall response file (uninstallresponse.txt) are located in the setup CD root directory. Important: Do not place the response file in a path that contains a space and do not put a space in the file name.</td>
</tr>
</tbody>
</table>

**Important:** If you have a Solaris 9 on Sun SPARC, you MUST either add the -W defaults.force32bit=true parameter to your installation command or install the 32-bit WebSphere Application Server product before installing WebSphere Portal to prevent the installer from automatically selecting a 64-bit environment.

**Note:** If the installation program does not detect a WebSphere Application Server instance that you know exists, exit the installation program and pass the WebSphere Application Server instance location using the command line; for example, ./install.sh -W was.undetectedWas="/my/WAS/location".

**Note:** If the GUI or console mode installation program fails to detect ports for either WebSphere Application Server or WebSphere Portal, a warning message displays and the installer offers another chance to enter the values. If the silent installation fails to detect ports for either WebSphere Application Server or WebSphere Portal, the installer will exit.

6. Verify your installation was successful; access WebSphere Portal using the http://yourserver:yourport/wps/portal format. Run the .//ConfigEngine.sh list-server-ports -DWasPassword=password task from the wp_profile_root/ConfigEngine directory to generate the wp_profile_root/ConfigEngine/log/wp_PortMatrix.txt file that lists WebSphere Portal ports for your installation.

7. Optional: After the WebSphere Portal installation completes successfully, run the .//ConfigEngine.sh configure-express -DPortalAdminPwd=password -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, if you want to enable the sample IBM Lotus Web Content Management content, such as internet and intranet sample sites.

- **Restriction:** Run the configure-express task before configuring your database, user registry, context root, security, etc. If you ran any tasks other than the install task, do not run this task.

**Note:** See http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_ovw.html
for tutorials on how to use the sample content.

The sample content includes:

- Creates a group called contentAuthors; members of this group are given privileges to create content in the sample Internet and intranet sites. Navigate to the Administration area and then click Access > Users and Groups.
- Creates two new Web Content Management Libraries: "Internet Web Content 6.1.0" and "Intranet Web Content 6.1.0". Navigate to the Administration area and then click Portal Content > Web Content Libraries.
- Adds a portlet filter and applies the filter to various portlets in the sample Internet and intranet sites. You can see the definition of the filter in the WebSphere Application Server Administration console and examining the custom resources under the Environment area.
- Creates two new theme policies: InternetStyle and IntranetStyle. These styles are applied to sample Internet and intranet sites. Navigate to Theme Customizer and then select the style.
- Creates several portlet clones of the Web Content Management rendering portlet. These portlet clones are used on sample Internet and intranet sites.
- Creates two virtual portals with context roots of wps/portal/intranet and wps/portal/internet. These are the sample Internet and intranet sites. Go to http://yourserver:yourport/wps/portal/internet and http://yourserver:yourport/wps/portal/intranet to access them.
- Creates several sample credential slots, including "Default slot for E-mail", "Default slot for Feeds", "Default slot for Miscellaneous", "Default slot for Web Clipping", and "Default slot for Web Content Management". Navigate to the Administration area and then click Access > Credential Vault > Manage System Vault Slots.
- Modifies the portlet palette to include some new portlets. Click the Portlets link to see the portlet palette.

8. Optional: If you ran the configure-express task, the owner of the items in the Web content libraries containing the Internet and Intranet Site Template content will be listed as uid=xyzadmin,o=defaultWIMFileBasedRealm. To update the owner information for these items to correspond to your portal administrator ID, complete the following steps:

A. Edit the wp_profile_root
   /PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties file.
B. Add the following line to the file:
   
   uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
   
   Replace portal_admin_DN with the distinguished name of the portal administrator.
C. Save your changes and close the file.
D. Run the ./ConfigEngine.sh action-express-memberfixer -DPortalAdminPwd=password -DWasPassword=password -DModifyPortsServer=servername -DStartPort=starting port number task, located in the wp_profile_root/ConfigEngine directory.

9. Optional: Perform the following steps if you need to change the ports for WebSphere Application Server or WebSphere Portal:

   **Note:** The starting port parameter is required for a successful completion of the modify-ports-by-startport task.

   Once you specify a start port, this port becomes the base for assigning port values. The code increments this value as each port is assigned, which means that the WebSphere Portal ports will be assigned incrementally starting with the port defined with the modify-ports-by-startport task.

   A. Stop the server1 and WebSphere_Portal servers.
   B. Run one of the following commands for each server you need to change:

<table>
<thead>
<tr>
<th>Method</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting port number</td>
<td>./ConfigEngine.sh modify-ports-by-startport -DWasPassword=password -DModifyPortsServer=servername -DStartPort=starting port number</td>
</tr>
</tbody>
</table>
C. Restart the server1 and WebSphere_Portal servers.

10. To prevent Out of Memory errors, perform the following steps to set the MaxPermSize:

   Tip: If Maximum Heap Size is set to 2048 M or higher, set the MaxPermSize to a quarter of the value entered for the Maximum Heap Size; for example, if your Maximum Heap Size is 3000 M, set MaxPermSize to 750 M. If your Maximum Heap Size is less than 2048 M, set MaxPermSize to 512 M.

   A. Log in to the WebSphere Application Server Administration Console.
   B. Click Servers > Server Types > WebSphere application servers > WebSphere_Portal.
   D. In the Generic JVM arguments field, change the MaxPermSize value to -XX:MaxPermSize=numerical value, where numerical value is a quarter of the value entered for the Maximum Heap Size. Important: If MaxPermSize does not exist in the Generic JVM arguments field, add it to the field but do not replace existing information in the Generic JVM arguments field with the MaxPermSize information.
   E. Click OK to save your changes.
   F. Click Save to save your changes to the master configuration.
   G. Log out of the WebSphere Application Server Administration Console.
   H. Restart the server1 and WebSphere_Portal servers.

Parent topic: Setting up a stand-alone server on Solaris
Previous topic: Preparing your Solaris operating system
Next topic: Configuring databases

Related concepts
Installation methods
IBM Support Assistant Lite for WebSphere Portal

Related tasks
Creating and maintaining multiple profiles on Solaris
Using the member fixer tool with IBM Lotus Web Content Management

Related reference
Port file Note: Sample port files are available on the Setup disc.

Note:
Sample port files are available on the Setup disc.

./ConfigEngine.sh modify-ports-by-portsfile -DWasPassword=password -DModifyPortsServer=servername -DPortsFile=full path to ports file

The following is an example of the information within a port file although the port values will be different based on your environment:

BOOTSTRAP_ADDRESS=10031
SOAP_CONNECTOR_ADDRESS=10032
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=10032
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=10025
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=10036
WC_adminhost=10027
WC_defaulthost=33344
DCS_UNICAST_ADDRESS=10029
WC_adminhost_secure=10039
WC_defaulthost_secure=10035
SIB_ENDPOINT_ADDRESS=10026
SIB_ENDPOINT_SECURE_ADDRESS=10037
SIB_MQ_ENDPOINT_ADDRESS=10030
SIB_MQ_ENDPOINT_SECURE_ADDRESS=10028
ORB_LISTENER_ADDRESS=10034
Advanced installation parameters
Configuring databases
Before transferring default data to the production database server, you need to prepare the database server. Database preparation includes creating user IDs and databases. For this installation path, stand-alone production server, remote databases servers are used.
Select the database server that are using.

- Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server
  You can install DB2 on the same server as WebSphere Portal, also referred to as local, or on a different server, referred to as remote. If you plan to use a local DB2 database, scripts are provided to simplify the database creation and transfer. More information about the local database is provided in the single server installation instructions. For a remote database you must complete the transfer manually.

- Preparing DB2 for z/OS for a stand-alone production server
  Setting up the DB2 for z/OS database includes creating user IDs, databases, and table spaces on a remote server.

- Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server
  Setting up the Oracle database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

- Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server
  Setting up the Oracle RAC database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

- Setting up a remote SQL Server database on Windows for a stand-alone production server
  To setup a remote Microsoft SQL Server Enterprise Edition database, create user IDs and databases on a remote server. Tasks are provided to assist with creating the user IDs and the databases. Before you can use the tasks, you must modify properties files.

- Verifying databases
  After you configure IBM WebSphere Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.

Parent topic: Setting up a stand-alone server on Solaris
Previous topic: Installing WebSphere Portal on Solaris
Next topic: Preparing a remote Web server when portal is installed on Solaris
Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

You can install DB2 on the same server as WebSphere Portal, also referred to as local, or on a different server, referred to as remote. If you plan to use a local DB2 database, scripts are provided to simplify the database creation and transfer. More information about the local database is provided in the single server installation instructions. For a remote database you must complete the transfer manually.

1. **Installing DB2**
   View information on installing DB2 for use with WebSphere Portal.

2. **Create users**
   View the steps to set up users for DB2 to work with WebSphere Portal.

3. **Creating remote databases**
   A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

4. **Optional: Assigning custom table spaces**
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

5. **Modifying database properties**
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

6. **Optional: Configuring JCR collation support**
   View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

7. **Configure WebSphere Portal to use DB2**
   View the steps to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

8. **Configuring DB2 for large file handling in Web Content Management**
   If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

9. **Optional: Changing driver types**
   WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

**Parent topic:** Configuring databases
Installing DB2

View information on installing DB2 for use with WebSphere Portal.

Before you begin:

- Make sure DB2 is set up with updated kernel parameters according to the recommendations in the DB2 Quick Beginnings guide at DB2 Technical Support.
- **(For Type 2 drivers only):** DB2 Connect must be installed on the same system as WebSphere Portal, and have the same name as the server profile name.
- When you install DB2 using the DB2 installation program, it automatically creates a DB2 administrative user with the correct operating system rights.

1. If you are using the JDBC driver in type 2 mode, configure your DB2 Connect client with the following commands. If you are using a remote database, complete this step separately from the WebSphere Portal installation.

```bash
- db2 update dbm cfg using tp_mon_name WAS
- db2 update dbm cfg using spm_name hostname, where hostname is the host name of WebSphere Portal.
```

Because the default for `spm_name` is the hostname itself, specifying the `hostname` parameter is optional. If your hostname is more than eight characters, use empty double quotes (" "). For example, `db2 update dbm cfg using spm_name " "`.

2. Before installing DB2, log in with a user ID that has administrative authority. The system user ID and password must match the database user ID and password for the WebSphere Portal databases. If you do not use the default DB2 database user ID, or if you must access a remote database, create the system user ID before installing DB2.

To set up the environment for the database access:

A. The initialization script for this user (for example, `user-home/.profile`) must contain a call to the `db2profile` script in the `db-home/sqllib` directory.

B. After you create the system user ID and password for the DB2 installation, add the user ID to the DB2 administration group (such as `db2admn`) for that system.

To ensure that the user is set up correctly, login with this user ID and start the DB2 command line processor by executing the command: `db2`. If the command line processor displays, the user ID is set up correctly.

3. To install DB2, follow the instructions that are provided with the DB2 documentation to install DB2 and the required fix pack.

**Parent topic:** Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

**Next topic:** Create users
Create users

View the steps to set up users for DB2 to work with WebSphere Portal.
Before you begin: You should have completed Installing DB2.

- Prerequisites
  - Installing DB2

For all databases, use one user with administrative rights on the operating system and the DB2 installation. This user can be the database administrative user that is created automatically by the DB2 installation program. This user is the database configuration user that will be used for configuration tasks: creating database tables and performing database transfer. If you choose to have WebSphere Portal also create the databases, the database user should be given SYSADM rights. You only need to manually create an administrator ID when you do not want to use an existing DB2 administrator ID.

A common user name is `db2inst1`, but you can assign any user name as long as it has administrative access and follows the limitations listed here. Do not change the user name after creating it.

The limitations on user names are:
- User names can contain one to eight characters.
- Group and instance names can contain one to eight characters.
- Names cannot be any of the following: users, admins, guests, public, local
- Names cannot begin with: IBM, SQL, SYS
- Names cannot include accented characters.
- Create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Following are the basic steps to create new users and grant privileges. For more information, refer to the DB2 product documentation.

1. From the DB2 Control Center, expand the object tree until you find the DB Users folder.
2. Right-click the DB Users folder and select Create/Add. The Create DB User notebook opens.
3. In the User name field, type a user ID.
4. Assign an authority for the user.
5. Click OK.

Parent topic: Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server
Previous topic: Installing DB2
Next topic: Creating remote databases
Creating remote databases

A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

Before creating these databases, note the following information:

- Use IBM® DB2 Universal Database™ Enterprise Server Edition JDBC driver, JCC, type 4.
- If the DB2 JDBC type 4 driver is used, only the instructions on the remote server are required. You do not need to install the DB2 client software, and you do not need to complete the steps related to the DB2 client.
- The client software, DB2 Connect, must be correctly configured to connect to the remote DB2 server instance, for example, `db2inst1`.
- These instructions assume that a remote DB2 server and DB2 Connect are already installed and running. For information on installing DB2 Connect, see the DB2 product documentation.

**Prerequisites**

- Installing DB2
- Create users

To create a database, you must be a DB2 System Administrator with sufficient database privileges (SYSADM or at a minimum SYSCTRL).

1. Log in to the DB2 server system as a user with sufficient database privileges.
2. Initialize a DB2 command environment by opening a command prompt and typing `su - db2inst1`, where `db2inst1` is the database user. The prompt changes to your operating system shell prompt, for example: `$`. In this mode, you must type `db2` at the beginning of each command and use double quotation marks (""") around the entire command. The following example shows how to use the double quotes; it is not an actual command that you run:

   ```
   db2 "CREATE REGULAR TABLESPACE SMS PAGESIZE 4 K MANAGED BY SYSTEM USING ('sms') BUFFERPOOL SMSPOOL"
   ```

   **Note:** If you are using the Command Line Processor (CLP), refer to your DB2 documentation for details. The command prompt is `db2=>`.

3. Run the following commands on the DB2 server system to configure the DB2 database instance:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>db2set DB2_RR_TO_RS=YES</code></td>
</tr>
<tr>
<td></td>
<td><code>db2set DB2_EVALUNCOMMITTED=YES</code></td>
</tr>
<tr>
<td></td>
<td><code>db2set DB2_INLIST_TO_NLJN=YES</code></td>
</tr>
<tr>
<td></td>
<td><code>db2 &quot;UPDATE DBM CFG USING query_heap_sz 32768&quot;</code></td>
</tr>
<tr>
<td></td>
<td><code>db2 &quot;UPDATE DBM CFG USING maxagents 500&quot;</code></td>
</tr>
<tr>
<td></td>
<td><code>db2 &quot;UPDATE DBM CFG USING sheapthres 0&quot;</code></td>
</tr>
</tbody>
</table>

4. **(Important)** Failure to complete this step will result in an unsuccessful database transfer. Run the following commands on the DB2 server system to create the necessary databases:

   ```
   Notes:
   ```

   - Replace `dbname` with the actual name of the database. Run the commands and each time replace `dbname` with the actual values for release, community, customization, Java Content Repository, Feedback, and Likeminds. You will need to run the commands once for each database for a total of six times.

   **Remember:** DB2 database names cannot...
exceed eight characters. Therefore, consider using these database names: release, commun, custom, jcrdb, fdbkdb, and lmdb.

```
db2 "CREATE DB dbname using codeset UTF-8 territory us PAGESIZE 8192"
db2 "UPDATE DB CFG FOR dbname USING applheapsz 4096"
db2 "UPDATE DB CFG FOR dbname USING app_ctl_heap_sz 1024"
db2 "UPDATE DB CFG FOR dbname USING stmtheap 32768"
db2 "UPDATE DB CFG FOR dbname USING dbheap 2400"
db2 "UPDATE DB CFG FOR dbname USING locklist 1000"
db2 "UPDATE DB CFG FOR dbname USING logfilsiz 4000"
db2 "UPDATE DB CFG FOR dbname USING logprimary 12"
db2 "UPDATE DB CFG FOR dbname USING logsecond 20"
db2 "UPDATE DB CFG FOR dbname USING logbufsz 32"
db2 "UPDATE DB CFG FOR dbname USING avg_appls 5"
db2 "UPDATE DB CFG FOR dbname using AUTO_MAINT off"
```

Complete the following:

5. On the DB2 server system, run the following commands. This step is only required for the IBM Java Content Repository database (jcrdb).

- `jcrdb` is the name of the database used to store user data and objects
- `jcr` is the jcr user for `jcrdb`
- `dbpassword` is the password for the jcr user for the `jcrdb`

```
db2 "CONNECT TO jcrdb USER jcr USING dbpassword"
db2 "CREATE BUFFERPOOL ICMLSFRQBP4 SIZE 1000 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL ICMLSVOLATILEBP4 SIZE 16000 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL ICMLSMAINBP32 SIZE 16000 PAGESIZE 32 K"
db2 "CREATE BUFFERPOOL CMBMAIN4 SIZE 1000 PAGESIZE 4 K"
```

For JDBC Type 2 connections only: On DB2 Connect, use a text editor to open the `/etc/services` file. If it does not specify the DB2 connection service port, add the following text to specify the port for the remote DB2 instance:

```
db2 "DISCONNECT jcrdb"
db2 "TERMINATE"
```

B. For JDBC Type 2 connections only: On DB2 Connect, use a text editor to open the `/etc/services` file. If it does not specify the DB2 connection service port, add the following text to specify the port for the remote DB2 instance:

```
db2inst1port1/tcp # DB2 connection service port
```

where `db2inst1` is the name of the DB2 instance ID on the system, and `port1` with the actual port number that is
assigned to the DB2 connection service in your DB2 server installation. The connection service port on the DB2
Client system, WebSphere Portal server, must match the connection service port on the DB2 server. The ports
should match by number but not necessarily by name.

C. Set up the correct service name by entering the following command on the DB2 server system:
   `db2 "UPDATE DBM CFG USING svcename svce_name"` where `svce_name` is the connection service port name that is specified above.

D. For JDBC Type 2 connections only: On DB2 Connect, set DB2COMM to TCP/IP by using the db2set command
   `db2set DB2COMM=tcpip`.

E. For JDBC Type 2 connections only: Catalog the TCP/IP node with the IP address of the remote database server on
   DB2 Connect:
   `db2 "catalog tcpip node remote_db_node_alias remote database_server_node server connection_service_port"` where:
   - `remote_db_node_alias` is the alias name of the database server that you are defining for the WebSphere Application
     Server node name. The alias name can contain one to eight characters.
   - `database_server_node` is the fully qualified host name of your database server system.
   - `connection_service_port` is the name of the DB2 connection service port that is configured in the `/etc/services` file on the database server system.

F. For JDBC Type 2 connections only, catalog the WebSphere Portal databases on DB2 Connect, where:
   - `remote_db_name_domain`, is the cataloged name of the databases on the server system for each domain.
   - `domain_alias_name`, is the database alias names that you are defining.
   - `remote_db_node_alias` is the name that was used previously when you cataloged the TCP/IP node in the previous
     step.

   The alias for each database must be different from the actual database name and can only contain up to eight
   characters.
   `db2 "catalog db remote_db_name_release as release_alias_name at node remote_db_node_alias"

   `db2 "catalog db remote_db_name_community as comm_alias_name at node remote_db_node_alias"

   `db2 "catalog db remote_db_name_customization as cust_alias_name at node remote_db_node_alias"

   `db2 "catalog db remote_db_name_fdbkdb as fdbkdb_alias_name at node remote_db_node_alias"

   `db2 "catalog db remote_db_name_lmdb as lmdb_alias_name at node remote_db_node_alias"

   `db2 "catalog db remote_db_name_jcrdb as jcrdb_alias_name at node remote_db_node_alias"

G. For JDBC Type 2 connections only: Log out of DB2 Connect by entering `db2 "terminate"`.

H. For JDBC Type 2 connections only, on DB2 Connect, test your remote connection by issuing the following command
   in the DB2 command window: `db2 "connect to alias_name user username using password"`, where `alias_name`
   is the alias name that you defined above, `username` is the database user, and `password` is the password assigned to
   the database user.

---

**Parent topic:** Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Create users

**Next topic:** Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the wp_profile_root /PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- The page size of table spaces used by WebSphere Portal must be 8192 bytes.
- For details on creating table spaces refer to the documentation for the database.

Prerequisites
- Installing DB2
- Create users
- Creating remote databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.index_name.indexspace
   For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword IN and a space. For example: community.COMP_INST.tablespace=IN COMM8KSPACE
   Repeat this step for each domain that you are transferring.
4. Save and close dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
   - UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

**Parent topic:** Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server  
**Previous topic:** Creating remote databases  
**Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

Tip: For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Working with properties files:

- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:
  - release.DbName=release
  - jcr.DbName=jcrdb
  - feedback.DbName=fdbkdb
  - likeminds.DbName=lmdb
  - community.DbName=commdb
  - customization.DbName=custdb

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.

- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds

- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
- \textit{dbdomain}.DbType
- \textit{dbdomain}.DbName
- \textit{dbdomain}.DbUrl
- \textit{dbdomain}.DbSchema

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If \textit{DbUser}, \textit{DbUrl}, and \textit{DbPassword} are not the same across domains, the value for \textit{DataSourceName} must differ from the \textit{DataSourceName} of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   - \texttt{wp_profile_root}/ConfigEngine/properties/wkplc.properties
   - \texttt{wp_profile_root}/ConfigEngine/properties/wkplc_comp.properties
   - \texttt{wp_profile_root}/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In \texttt{wkplc_comp.properties}, most properties are repeated for each domain.

2. Use a text editor to open the properties file \texttt{wkplc_comp.properties} and modify the values to correspond to your environment.
   A. For \textit{dbdomain}.DbType, type \texttt{db2}.
   B. For \textit{dbdomain}.DbName, type the name of the WebSphere Portal domain database. \textbf{Notes:}
      - This value is also the database element in the \textit{dbdomain}.DbUrl property.
      - This value is the TCP-IP alias for the database.
   C. For \textit{dbdomain}.DbSchema, type the schema name of the database domain. \textbf{Note:} Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   D. For \textit{dbdomain}.DataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For \textit{dbdomain}.DbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. \textbf{Note:} The database element of this value should match the value of \textit{DbName}.
   F. For \textit{dbdomain}.DbUser, type the user ID for the database administrator.
   G. For \textit{dbdomain}.DbPassword, type the password for the database administrator.
   H. Optional: For \textit{dbdomain}.DbRuntimeUser, type the user ID of the database user that should be used by WebSphere Portal to connect to the database at runtime.
   I. If \textit{dbdomain}.DbRuntimeUser is specified, you must set \textit{dbdomain}.DbRuntimePassword to be the password of the runtime database user.
   J. For \textit{dbdomain}.DBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.
K. For dbdomain.DBA.DbPassword, type the database administrator password for privileged access operations during creation of the database.

L. For dbdomain.XDbName, type the database loop back alias that needs to be set if you plan to use the create-database task. **Note:** Required only for local databases using a JDBC Type 2 driver.

M. For dbdomain.DbNode, type the value for the node database. Set this value if you want to call create-database. **Note:** Required only for local databases.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For source.domainDbType, type of the database you are currently configured to use. The value for source.domainDbType is Derby by default.

B. For source.domainDbName, type the name of the database domain you are currently using.

C. For source.domainDbSchema, type current schema identifier for objects within the database for this domain.

D. For source.domainDataSourceName, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For source.domainDbUrl, type the url currently used to access your database.

F. For source.domainDbUser, type the name of the user accessing this database.

G. For source.domainDbPassword, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file wkplc_dbtype.properties.

   A. For db2.DbDriver, type the name of the JDBC driver class.

   B. For db2.DbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For db2.JdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file wkplc.properties.

   A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server  
**Previous topic:** Assigning custom table spaces  
**Next topic:** Configuring JCR collation support
Configuring JCR collation support

View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties

1. Stop the WebSphere Portal server:
   - `./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password`

2. If using a remote DB2 server, copy the following files from the WebSphere Portal server to a temporary directory on the remote DB2 server:
   - `PortalServer_root/jcr/prereq.jcr/config/collation.jar`
   - `PortalServer_root/jcr/prereq.jcr/config/registerCollationUDFTemplate.sql`

3. Set up collation on the database where the JCR domain is located.
   A. Change to the directory `db2_instance_owner_home/sql/lib/function`.
   B. Execute the command: `Remote DB2:db2_instance_owner_home/sql/lib/java/jdk/bin/jar -xvf temporary location/collation.jar icm/CollationUDF.class`
   - `Local DB2:db2_instance_owner_home/sql/lib/java/jdk/bin/jar -xvf PortalServer_root/jcr/prereq.jcr/config/collation.jar icm/CollationUDF.class`
   C. If using a local DB2, change to the directory `wp_profile_root/PortalServer/jcr/config`. If using a remote DB2, change to the temporary directory where you copied the files in a previous step.
   D. Open the file `registerCollationUDFTemplate.sql` and change all `SCHEMA` references to the JCR schema; for example, `JCR`. The value set for `SCHEMA` should match the value set for the `jcr.DbSchema` property. You specify `jcr.DbSchema` in the configuration file `wkplc_comp.properties` when you modify database properties.
   E. Connect to the JCR database by running `db2 connect to <jcrdb> user <userid> using <password>`. You must connect to the JCR database using the userid and password of the WebSphere Portal JCR data source.
   F. Execute the script by running the command: `Remote DB2:db2 -tvf temporary location/registerCollationUDFTemplate.sql`
   - `Local DB2:db2 -tvf wp_profile_root/PortalServer/jcr/config/registerCollationUDFTemplate.sql`
   G. Disconnect from the JCR database.
   H. Restart the DB2 instance.

4. Verify that the UDF is registered properly.
   A. Log in as the db2instanceID. Open a DB2 terminal window, and type `db2`. From the command line, type the command`connect to JCRDB user userid using password`. You must connect to the JCR database using the userid and password of the WebSphere Portal JCR data source.
   B. Execute the following command in the previous substep: . If the command completes successfully, the UDF is registered correctly as follows: `values schema.sortkeyj('abc','en')`, where `schema` is the schema used in the previous substep.

5. Edit the `icm.properties` file, located in `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm` directory. Add the following section to the end of the file: `# Enable/Disable collation support for all DB2 platforms`
# Disabled by default
jcr.query.collation.db2.enabled = true

# Database specific collation mappings
# These mappings apply map a Java locale name into a collation name
# supported by the underlying database.
# Example mappings for DB2 platform

# English
jcr.query.collation.en = en

# Swedish
jcr.query.collation.sv = sv

jcr.query.collation.zh = zh
jcr.query.collation.de = de
jcr.query.collation.da = da
jcr.query.collation.hu = hu
jcr.query.collation.jp = jp

6. Enter the following command to start the WebSphere Portal server:

```
./startServer.sh WebSphere_Portal
```

**Parent topic:** Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server  
**Previous topic:** Modifying database properties  
**Next topic:** Configure WebSphere Portal to use DB2
Configure WebSphere Portal to use DB2

View the steps to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties

Tips:
- To run these tasks as a non-root user, you must first run the task chown -R non-root_user WebSphereDir.
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

Steps for transferring data to another supported database

1. If you are running a type 2 connection, before transferring data edit the db2cli.ini file. Failure to follow these steps will cause the database transfer to hang at the task action-process-constraints.
   A. Locate the file /home/db2inst1/sqllib/cfg/db2cli.ini.
   B. Add the following to the end of the file. Leave an empty line after ReturnAliases=0.[COMMON]

   
   [COMMON]

   DYNAMIC=1
   ReturnAliases=0

2. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.

3. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2.

   /home/db2inst1/sqllib/db2profile

   where db2inst1 represents your database instance Note: You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.

4. Enter the following commands to validate configuration properties...

   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password

5. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.
6. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td><code>.stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td><code>.stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

7. Transfer the database:

**Note:** Do not execute the database-transfer task as a background process. This might cause the task to stall.

A. Change to the directory `wp_profile_root/ConfigEngine`.
B. Enter the following command:

```
./ConfigEngine.sh database-transfer -DWasPassword=password
```

**Note:**
- To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:

```
./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
```

- If you have been storing data in IBM® Derby for a long time, database transfer could fail with OutOfMemory exceptions. If database transfer fails, add the following property to the command in this step:

```
./ConfigEngine.sh database-transfer -DDbtJavaMaxMemory=1536M -DWasPassword=password
```

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

8. After transferring the database tables, perform a `reorg` check to improve performance. Perform this step for each database alias in the property file.

A. Connect to a database with the following command:

```
db2 connect to database_alias user db2admin_userid using password
```

**Note:** Additional options might be required if additional security has been installed. Refer to DB2 Universal Database commands by example for links to the command reference.

B. After it is connected, run the following commands from the DB2 prompt:

```

db2 reorgchk update statistics on table all > xyz.out
```

C. Look in the `reorg` column for entries marked with a * (star or asterisk) in the file `xyz.out`. For each line with a *, note the `tablename` and run the following command for each `tablename`:

```
db2 reorg table tablename db2 terminate db2rbind
database_name -l db2rbind.out -u db2_admin -p password
```

D. The output file `db2rbind.out` is only created when there is an error for the `db2rbind` command.

9. Change to the directory `wp_profile_root/bin`.

10. Enter the following command to start the WebSphere Portal server:

```
./startServer.sh WebSphere_Portal
```

Parent topic: Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Configuring JCR collation support

Next topic: Configuring DB2 for large file handling in Web Content Management
Configuring DB2 for large file handling in Web Content Management

If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

**Prerequisites**
- Installing DB2
- Create users
- Creating remote databases
- Modifying database properties
- Configure WebSphere Portal to use DB2

**Note:** You only need to perform these steps if you are using Web Content Management.

1. Log into the WebSphere Application Server administrative console.
2. Click **Resources > JDBC > Data sources**.
3. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
4. Select the name of the data source that is defined in `wkplc_comp.properties` for the JCR database domain. The default data source is `wpdbDS`.
5. Click **Custom properties**.
6. Ensure that the `fullyMaterializeLobData` property is set to `false`.

**Parent topic:** Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Configure WebSphere Portal to use DB2

**Next topic:** Changing driver types
Changing driver types

WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Before you begin, ensure that the following conditions are met:
- The WebSphere Portal database has been successfully transferred to DB2 using the `database-transfer` configuration task.
- The files `wkplc_comp.properties` and `wkplc_dbtype.properties` have been modified to set the correct values for the DB2 drivers that you are switching to:

```sh
In the file `wkplc_comp.properties` set each `<Domain>.DbUrl` property using the following formats:
- db2 (type 2):        { jdbc:db2:wpsdb }
- db2 (type 4):        { jdbc:db2://<YourDatabaseServer>:50000/wpsdb:returnAlias=0; }
```

```sh
In the file `wkplc_dbtype.properties` set the `db2.DbLibrary` property using the following format:
- For DB2 Type 2 driver use `<SQLLIB>/java/db2java.zip`
- For DB2 Type 4 driver use `<SQLLIB>/java/db2jcc.jar:<SQLLIB>/java/db2jcc_license_cu.jar`
```

```sh
In the file `wkplc_dbtype.properties` set the `db2.DbDriver` property using the following format:
- For DB2 Type 2 driver use `COM.ibm.db2.jdbc.app.DB2Driver`
- For DB2 Type 4 driver use `com.ibm.db2.jcc.DB2Driver`
```

**Note:** If WebSphere Portal is installed on the same machine as the DB2 server and you switch from a JDBC Type 4 connection to a JDBC Type 2 connection, verify that you have created the alias names for the DB2 databases as described in Creating remote databases and that the alias names are specified for the databases in the file `wkplc_comp.properties`. When switching from a JDBC Type 2 connection to a JDBC Type 4 connection, remove the database alias names and refer to the databases directly. This is required because of a limitation in the DB2 Universal JDBC driver.

**Prerequisites**
- Installing DB2
- Create users
- Creating remote databases
- Modifying database properties
- Configure WebSphere Portal to use DB2
- Configuring DB2 for large file handling in Web Content Management

1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.
2. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2. 

   ```sh
   /home/db2inst1/sqlib/db2profile
   ```

   where `db2inst1` represents your database instance

   **Note:** You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.

3. Enter the following commands to validate configuration properties:

   ```sh
   /home/db2inst1/sqlib/db2profile
   ```

   ```sh
   DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ```

   **Note:** You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.
4. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td><code>./stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td><code>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

6. Change to the directory `wp_profile_root/ConfigEngine`.

7. To change from one supported driver to the other, run the following task to connect the database, including only the domains that require the switch. `./ConfigEngine.sh connect-database -Drelease.DbPassword=password -Dcustomization.DbPassword=password -Dcommunity.DbPassword=password -Djcr.DbPassword=password -Dfeedback.DbPassword=password -Dlikeminds.DbPassword=password -DWasPassword=password`

8. Change to the directory `wp_profile_root/bin`.

9. Enter the following command to start the WebSphere Portal server: `./startServer.sh WebSphere_Portal`

Parent topic: Preparing DB2 on HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Configuring DB2 for large file handling in Web Content Management
Preventing DB2 for Z/OS for a stand-alone production server

Setting up the DB2 for z/OS database includes creating user IDs, databases, and table spaces on a remote server.

1. Installing DB2 for z/OS
   View information on how to install DB2 for z/OS for use with IBM® WebSphere® Portal.

2. Creating users
   Complete this procedure to set up users for IBM DB2 Universal Database™ for z/OS® to work with IBM WebSphere Portal.

3. Creating remote databases
   A remote database resides on a different machine than WebSphere Portal. You must manually create the required databases before configuring WebSphere Portal to work with DB2 for z/OS.

4. Creating the Java Content Repository database
   View information on setting up your Java Content Repository database to work with WebSphere Portal.

5. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

6. Modifying database properties
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

7. Configuring WebSphere Portal to use DB2 for z/OS
   View information on manually transferring data to the DB2 for z/OS you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository to DB2 for z/OS. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

8. Verifying database connections
   After WebSphere Portal is configured to work with your database, test the database connection to ensure that it operates correctly.

Parent topic: Configuring databases
Installing DB2 for z/OS

View information on how to install DB2 for z/OS for use with IBM® WebSphere® Portal.

The following prerequisites must exist separately from the WebSphere Portal installation:

- If you are using the DB2 Legacy Type 2 JDBC driver, configure your DB2 Connect client with the following commands:
  - `db2 update dbm cfg using tp_mon_name WAS`
  - `db2 update dbm cfg using spm_name hostname`, where `hostname` is the host name for the machine where the DB2 Connect client is installed (same as portal machine).

- If you are using the DB2 Legacy Type 2 JDBC driver, enable extended shared memory usage with the following commands:
  - `export EXTSHM=ON`
  - `db2set DB2ENVLIST=EXTSHM`
  - `db2start`

  To make the change permanent, you must add the environment variable by adding the following to the `profile.env` file:

  ```
  DB2ENVLIST='EXTSHM'
  ```

  in `/home/db2inst/sqllib/userprofile` add: `export EXTSHM=ON`

  **Note:** The shell must be reopened before restarting DB2 for z/OS.

- Both type 2 and type 4 drivers are supported. If you are using the DB2 Legacy Type 2 JDBC driver, the DB2 Connect client must be installed on the same machine as WebSphere Portal to connect to the remote database.

- The DB2 subsystem must be on a supported z/OS platform.

- Update the BP8K0 catalog bufferpool to 35,000 buffers before performing database transfer. You should change this value based on your environment. The SYSIBM.SYSDATABASE table resides in this bufferpool and is used extensively by DB2 for z/OS during database transfer.

To use DB2 for z/OS as the database software for WebSphere Portal, you must have DB2 for z/OS installed on your z/OS system. Refer to the DB2 for z/OS product documentation for instructions. Refer to the Planning for DB2 for z/OS topic for additional considerations for configuring DB2 for z/OS as the WebSphere Portal database.

**Parent topic:** Preparing DB2 for z/OS for a stand-alone production server

**Next topic:** Creating users
Creating users

Complete this procedure to set up users for IBM® DB2 Universal Database™ for z/OS® to work with IBM WebSphere® Portal.

Before you begin: You should have completed Installing DB2 for z/OS.

- **Prerequisites**
  - Installing DB2 for z/OS

- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.
- If you are configuring multiple WebSphere Portal instances to use a single DB2 for z/OS subsystem, you must create a unique database user for each WebSphere Portal instance. By default all database table names include the name of the database user used to access the data. Therefore, to prevent table name conflicts, you must create a unique database user for each WebSphere Portal instance on the shared DB2 for z/OS subsystem.
- Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Use the following steps to grant access permissions to the database users. Repeat these steps for each WebSphere Portal instance you are setting up.

1. Create all database user IDs in the security product you are using on z/OS. For jcrschema, the database schema name for IBM Java Content Repository data, create a group and connect it to the database user ID for Java Content Repository data, jcr. The following sample shows the RACF definition of such a user ID and group, where jcr is the database user ID for Java Content Repository data, yourDefaultUserGroup is your default RACF group for database user IDs, jcrschema is the database schema name for Java Content Repository data, and yourDefaultGroup is your default RACF group for groups. If you have some other security product such as Top Secret or ACF2 instead of RACF, translate the sample RACF definition into the appropriate syntax before running the job.

   ```
   ADDUSER jcr DFLTGRP(yourDefaultUserGroup) NAME('WAS DB2 ACCESS USER')
   PW USER(jcr) NOINTERVAL
   ALU jcr PASSWORD(********) NOEXPIRED
   ADDGROUP jcrschema SUPGROUP(yourDefaultGroup)
   CONNECT jcr GROUP(jcrschema)
   ```

2. Run the following SQL statements using a tool like SPUFI while logged on to the DB2 subsystem to grant appropriate rights on the newly created databases. If you are configuring multiple WebSphere Portal instances to use a single DB2 for z/OS subsystem, be sure to use the database user associated with the instance you are setting up. (C) create/alter tables

   ```
   (C) create/alter indice;
   (C+R) read/write data
   ```

   (C) - at configuration time
   (R) - at runtime
GRANT USE OF ALL BUFFERPOOLS TO releaseusr;
GRANT USE OF ALL BUFFERPOOLS TO communityusr;
GRANT USE OF ALL BUFFERPOOLS TO customizationusr;

GRANT DBADM ON DATABASE jcrdbnameonzos TO jcr;
GRANT USE OF ALL BUFFERPOOLS TO jcr;
GRANT DBADM ON DATABASE fdbkdbnameonzos TO feedback;
GRANT USE OF ALL BUFFERPOOLS TO feedback;
GRANT DBADM ON DATABASE lmdbnameonzos TO lmdbusr;
GRANT USE OF ALL BUFFERPOOLS TO lmdbusr;
GRANT SELECT ON SYIBM.SYSCOLUMNS TO releaseusr;
GRANT SELECT ON SYIBM.SYSCOLUMNS TO communityusr;
GRANT SELECT ON SYIBM.SYSCOLUMNS TO customizationusr;

GRANT SELECT ON SYIBM.SYSTABLES TO releaseusr;
GRANT SELECT ON SYIBM.SYSTABLES TO communityusr;
GRANT SELECT ON SYIBM.SYSTABLES TO customizationusr;

GRANT SELECT ON SYIBM.SYSCOLUMNS TO jcr;
GRANT SELECT ON SYIBM.SYSTABLES TO jcr;
GRANT SELECT ON SYIBM.SYSCOLUMNS TO fdbkdbus;
GRANT SELECT ON SYIBM.SYSTABLES TO fdbkdbus;
GRANT SELECT ON SYIBM.SYSCOLUMNS TO lmdbusr;
GRANT SELECT ON SYIBM.SYSTABLES TO lmdbusr;
GRANT SELECT ON SYIBM.SYSFOREIGNKEYS TO releaseusr;
GRANT SELECT ON SYIBM.SYSFOREIGNKEYS TO communityusr;
GRANT SELECT ON SYIBM.SYSFOREIGNKEYS TO customizationusr;

GRANT SELECT ON SYIBM.SYSFOREIGNKEYS TO jcr;
GRANT SELECT ON SYIBM.SYSFOREIGNKEYS TO fdbkdbus;
GRANT SELECT ON SYIBM.SYSFOREIGNKEYS TO lmdbusr;
GRANT SELECT ON SYIBM.SYSRELS TO releaseusr;
GRANT SELECT ON SYIBM.SYSRELS TO communityusr;
GRANT SELECT ON SYIBM.SYSRELS TO customizationusr;

GRANT SELECT ON SYIBM.SYSRELS TO jcr;
GRANT SELECT ON SYIBM.SYSRELS TO fdbkdbus;
GRANT SELECT ON SYIBM.SYSRELS TO lmdbusr;
GRANT USE OF STOGROUP jcrstogroup TO jcr;
GRANT CREATEIN, DROPIN ON SCHEMA jcrschema TO jcr;
GRANT SELECT ON SYIBM.SYSTABLESPACE TO jcr;
GRANT SELECT ON SYIBM.SYSVIEWS TO jcr;
GRANT SELECT ON SYIBM.SYSDUMMY1 TO jcr;
GRANT SELECT ON SYIBM.SYSTRIGGERS TO jcr;
GRANT SELECT ON SYIBM.SYSINDEXPART TO jcr;
GRANT SELECT ON SYIBM.SYSINDEXES TO jcr;
GRANT SELECT ON SYIBM.SYSSYNONYMS TO jcr;

where:
- releasenameonzos, communitynameonzos, customizationnameonzos, and releaseusr, communityusr, customizationusr represent the databases and database users, respectively, of the WebSphere Portal instance you are setting up. (These users must be created on the host system.)
- jcrdbnameonzos and jcr are the database and database user, respectively, for Content Repository data.
- fdbkdbnameonzos and feedback are the database and database user, respectively, for Feedback data.
- lmdbnameonzos and lmdbusr are the database and database user, respectively, for Likeminds data.
- jcrschema is the database schema name for Content Repository data.

3. Grant the necessary access rights to all users who might require them. Depending on the architecture that you choose, these users might include Java Content Repository and Feedback users.

**Parent topic:** Preparing DB2 for z/OS for a stand-alone production server  
**Previous topic:** Installing DB2 for z/OS  
**Next topic:** Creating remote databases
Creating remote databases

A remote database resides on a different machine than WebSphere Portal. You must manually create the required databases before configuring WebSphere Portal to work with DB2 for z/OS.

Before you begin, complete the following prerequisites:

- These instructions assume that you will use a local IBM® DB2 Universal Database™ Enterprise Server Edition Connect to connect to a remote IBM DB2 Universal Database for z/OS® server, and that IBM WebSphere® Application Server, WebSphere Portal, and DB2 Connect will be installed on the same machine.
- To run SQL statements, you can use a tool like SPUFI. The specified statements include CREATE STOGROUP, CREATE DATABASE and CREATE TABLESPACE. For CREATE STOGROUP you have to replace the `icmvolumes` and `icmvcat` variables with volume serial numbers and a catalog name for your environment. For CREATE TABLESPACE you can specify a specific BUFFERPOOL. Refer to the DB2 for z/OS SQL Reference for a description of these options.
- Ensure that a TEMP database is created for your subsystem. You can use the following statements to create a TEMP database:

```
CREATE DATABASE db_name AS TEMP;
CREATE TABLESPACE ts_name IN db_name;
```

- Log on to the DB2 subsystem on the host server. DB2 system administrator rights are needed to create the databases.
- Replace variables as follows:
  - `releasenameonzos`, `communitynameonzos`, and `customizationnameonzos` are the WebSphere Portal databases for WebSphere Portal and Member Manager data.
  - `fdbkdbnameonzos` and `fdbkdbts` are the database and table space, respectively, for Feedback data.
  - `lmdbnameonzos` and `lmdbts` are the database and table space, respectively, for LikeMinds data.

Because large objects are stored in columns that can become very large, logging changes to these columns requires a huge amount of log space. For this reason, large object (LOB) logging is disabled by default for tablespaces that contain such data. With LOB logging disabled, you can recover full backups, but not incremental backups that can be used for point in time recovery. To recover point in time backups, you must enable LOB logging. For detailed instructions, see Technote 1306637, Managing LOB logging in DB2 for z/OS.

**Prerequisites**

- Installing DB2 for z/OS
- Creating users

Use the following steps to set up WebSphere Portal with a remote instance of DB2 for z/OS. Refer to Planning for DB2 for z/OS for a list of databases and table space names. If you are configuring multiple WebSphere Portal instances to use a single DB2 subsystem, be sure to use the database and table space names associated with the WebSphere Portal instance you are setting up.

1. `CREATE DATABASE releasenameonzos CCSID UNICODE;`
2. `CREATE DATABASE communitynameonzos CCSID UNICODE;`
3. `CREATE DATABASE customizationnameonzos CCSID UNICODE;`
4. Execute the steps in the topic Creating the Java Content Repository database.
5. `CREATE DATABASE fdbkdbnameonzos CCSID UNICODE;`
6. `CREATE TABLESPACE fdbkdbts IN fdbkdbnameonzos USING STOGROUP SYSDEFLT PRIQTY 5000 SECQTY 500 LOCKSIZE ROW;`
7. `CREATE DATABASE lmdbnameonzos CCSID UNICODE;`
CREATE TABLESPACE lmdbts IN lmdbnameonzos USING STOGROUP SYSDEFLT PRIQTY 5000 SECQTY 500;
Creating the Java Content Repository database

View information on setting up your Java Content Repository database to work with WebSphere Portal. First, you need to create multiple databases for scalability.

The IBM Java Content Repository database grows in size as the IBM® WebSphere® Portal server is used. To support this growth, which includes the dynamic creation of many new tables within the database, the WebSphere Portal server allows you to create multiple databases within a single IBM DB2 Universal Database™ for z/OS® subsystem. A minimum of two databases is recommended, but more can be created if necessary. A single database setup is not recommended; it will quickly become constrained. See the Performance guides on the product documentation Web site for more information.

Each database that is created must begin with a common prefix; there is no convention required for the remainder of the name. For example, to create **x** number of databases, you may choose to use the following commands:

```
CREATE DATABASE JCRDB01
CREATE DATABASE JCRDB02
... 
CREATE DATABASE JCRDBxx
```

In this case, **JCR** is the common prefix. You also have to select a maximum number of user-defined tables (UDT) that each database will be allowed to hold; the recommended value is 400.

- **Prerequisites**
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases

A sample script of the commands that you can use to create the Java Content Repository database in your DB2 for z/OS installation is shown below. The sample demonstrates the creation of a single database. To follow the recommendation of creating at least two databases, duplicate the commands for each additional database as indicated below. Find a template of these commands in the file `PortalServer_root installer/wp.config/config/templates/db/db2_zos/jcr_sample.sql`.

**Notes:**

- It is not necessary to duplicate every tablespace definition in every database.
- In order to run the commands shown in the sample, you must know the database name and the IDs of the MVS volumes where the Java Content Repository database tables will be stored.
- Edit the template file for your installation environment before running the commands shown in the sample:
  - Replace `jcrdbnameX` with the name of the database. Remember that each database name must begin with a common prefix.
  - Replace `stogroup` with the name of your storage group.
  - Replace `icmvolumes` with the IDs of the MVS volumes where the Java Content Repository database tables will be stored. These values are assigned by the database administrator.
  - Replace `icmvcat` with the name of the virtual catalog.
  - Replace `jcr` with the name of database user ID.
  - Replace `4kbp` with the name of your 4K bufferpool.
  - Replace `32kbp` with the name of your 32K bufferpool.
  - Replace `jcrschema` with the schema name of your Java Content Repository domain.
--DROP DATABASE jcr dbnameX
--DROP STOGROUP stogroup
--DROP ALIAS jcrschema.ICMFICTITIOUS

CREATE STOGROUP stogroup VOLUMES (icmvolumes) VCAT icmvcat
GRANT USE OF STOGROUP stogroup to jcr

CREATE DATABASE jcr dbnameX STOGROUP stogroup BUFFERPOOL 4kbp INDEXBP 4kbp CCSID UNICODE

Note: The following two tablespace commands must be created in every database:

CREATE TABLESPACE ICMLFQ32 IN jcr dbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICMTSQ04 IN jcr dbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

Note: The following tablespaces are required in the first database only:

CREATE TABLESPACE ICMVFQ04 IN jcr dbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICMSFQ04 IN jcr dbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSTADO IN jcr dbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRIDLSTS IN jcr dbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRSCLSTS IN jcr dbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRISLSTS IN jcr dbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRGCLSTS IN jcr dbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRIGLSTS IN jcr dbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSTDPV IN jcr dbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCCLSTS IN jcr dbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSDOAC IN jcr dbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RESMLSTS IN jcr dbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COLNLSTS IN jcr dbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USERLSTS IN jcr dbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USEGLSTS IN jcr dbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCLLSTS IN jcr dbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE SYSCLSTS IN jcr dbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CACLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CPERLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COVALSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COMALSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COMDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CPERLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COVDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COMDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CACLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE IUC311 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE IUC321 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE IUT331 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT341 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT331 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT321 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT311 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT601 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE DWLSJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE GENDJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE GLBPJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE GENDJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE MAXSJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ADDPJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT601 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT341 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT331 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT321 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT311 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT601 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT341 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT331 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT321 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT311 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT601 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT341 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT331 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT321 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT311 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT601 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT341 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT331 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT321 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT311 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT601 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT341 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT331 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT321 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT311 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TSPEJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TSTAJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RIDSJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE WSNOJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE LOB TABLESPACE ICMLOBT1 IN jcrdbnameX LOG NO
CREATE LOB TABLESPACE ICMLOBT2 IN jcrdbnameX LOG NO
CREATE LOB TABLESPACE ICMLOBT3 IN jcrdbnameX LOG NO
CREATE ALIAS jcrschema.ICMFICTITIOUS FOR SYSIBM.SYSDUMMY1
CREATE TABLESPACE DLKRJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE LKRLJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.
- When using IBM® DB2 Universal Database™ for z/OS® as a data store, WebSphere Portal requires that its indexes are not padded. Therefore, you must set the DSNZPARM parameters to RETVLCFK=NO or PADIX=NO, or both.

- Prerequisites
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases
  - Creating the Java Content Repository database

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.index_name.indexspace
   For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
     - release
     - community
     - customization
     - jcr
     - feedback
     - likeminds
3. Assign a table space to each .tablespace entry in the mapping file. Assignments to .indexspace entries are ignored. The table space name must be qualified by the database name and prepended by the keyword IN and a space. For example: community.COMP_INST.tablespace=IN COMM8KSPACE
   Repeat this step for each domain that you are transferring.
4. Save and close dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

Parent topic: Preparing DB2 for z/OS for a stand-alone production server
Previous topic: Creating the Java Content Repository database
Next topic: Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases
  - Creating the Java Content Repository database

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:
  - `release.DbName=release`
  - `jcr.DbName=jcrdb`
  - `feedback.DbName=fdbkdb`
  - `likeminds.DbName=lmdb`
  - `community.DbName=commdb`
  - `customization.DbName=custdb`

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.

- Depending on which database domain has to be configured, replace `dbdomain` with:
  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
  - dbdomainDbType
  - dbdomainDbName
  - dbdomainDbUrl
  - dbdomainDbSchema

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

**Note:** To successfully transfer data from the JCR domain, you must use the DDF location value for the value of jcr.DbName field when setting up IBM DB2 Universal Database™ for z/OS®. You can locate the name of the DDF location value in the IBM DB2 Universal Database for z/OS sdsnsamp dataset, member DSNTIJUZ, or by running the following DB2 command:

```db2 subsystem prefix display ddf```

1. Locate the following files and create a backup copy of each before changing any values:
   - `<wp_profile_root>/ConfigEngine/properties/wkplc.properties`
   - `<wp_profile_root>/ConfigEngine/properties/wkplc_comp.properties`
   - `<wp_profile_root>/ConfigEngine/properties/wkplc_dbtype.properties`

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.
   A. For dbdomainDbType, type `db2_zos`.
   B. For dbdomainDbName, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the dbdomainDbUrl property.
   C. For dbdomainDbSchema, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   D. For dbdomainDbNameOnZos, type the name of the WebSphere Portal database on DB2 for z/OS. **Note:**
      - If running DB2 for z/OS as a remote database, set the value to the name of the remote database for the domain.
      - If WebSphere Portal is running on z/OS with DB2 for z/OS, set the value equal to the value of DbName.
   E. For dbdomainDataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   F. For dbdomainDbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:** The database element of this value should match the value of DbName.
G. For \texttt{dbdomain.DbUser}, type the user ID for the database administrator.

H. For \texttt{dbdomain.DbPassword}, type the password for the database administrator.

I. For \texttt{dbdomain.DbTablespace}, type the name of the DB2 for z/OS tablespace.

J. For \texttt{dbdomain.DbStorageGroup}, type the name of the storage group for the database.

K. For \texttt{dbdomain.DbVolumes}, type the volumes for the database.

L. For \texttt{dbdomain.DbVcat}, type the VCAT for the database.

M. For \texttt{dbdomain.Db4KBufferPoolName}, type the 4K bufferpool name for the database.

N. For \texttt{dbdomain.Db32KBufferPoolName}, type the 32K bufferpool name for the database.

O. For \texttt{dbdomain.DbIndex4KBufferPoolName}, type the 4K bufferpool name for the database. If you choose to use the default bufferpool value BP3, verify that this bufferpool is active.

P. For \texttt{dbdomain.TablespaceTrackMod}, set the value to determine TRACKMOD attribute of all tablespaces to use the specified value. Refer to the DB2 for z/OS documentation before changing this value.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in \texttt{wkplc_comp.properties}. \textbf{Important:} The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For \texttt{source.domainDbType}, type of the database you are currently configured to use. The value for \texttt{source.domainDbType} is derby by default.

B. For \texttt{source.domainDbName}, type the name of the database domain you are currently using.

C. For \texttt{source.domainDbSchema}, type current schema identifier for objects within the database for this domain.

D. For \texttt{source.domainDataSourceName}, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For \texttt{source.domainDbUrl}, type the url currently used to access your database.

F. For \texttt{source.domainDbUser}, type the name of the user accessing this database.

G. For \texttt{source.domainDbPassword}, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file \texttt{wkplc_dbtype.properties}.

A. For \texttt{db2_zos.DbDriver}, type the name of the class that \texttt{SqlProcessor} uses to import SQL files.

B. For \texttt{db2_zos.DbLibrary}, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

C. For \texttt{db2_zos.JdbcProviderName}, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

D. For \texttt{db2_zos.DbDriverType}, type the number of the driver type for the database.

E. For \texttt{db2_zos.DbLocationName}, type the DB2 location name. This value is set in the installation job DSNTIJUZ.

6. Save and close the file.

7. Update the following property in the file \texttt{wkplc.properties}.

A. For \texttt{WasPassword}, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

\textbf{Parent topic:} Preparing DB2 for z/OS for a stand-alone production server  
\textbf{Previous topic:} Assigning custom table spaces  
\textbf{Next topic:} Configuring WebSphere Portal to use DB2 for z/OS
Configuring WebSphere Portal to use DB2 for z/OS

View information on manually transferring data to the DB2 for z/OS you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository to DB2 for z/OS. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

- Prerequisites
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases
  - Creating the Java Content Repository database
  - Modifying database properties

Steps for transferring data to another supported database

1. If you are running a type 2 connection, before transferring data edit the db2cli.ini file. Failure to follow these steps will cause the database transfer to hang at the task action-process-constraints.
   A. Locate the file /home/db2inst1/sqllib/cfg/db2cli.ini.
   B. Add the following to the end of the file. Leave an empty line after ReturnAliases=0.[COMMON]
      
      DYNAMIC=1
      ReturnAliases=0

2. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.

3. Enter the following commands to validate configuration properties.
   
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password

4. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td>./stopServer.sh server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

6. Transfer the database:
   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory wp_profile_root/ConfigEngine.
B. Enter the following command: `./ConfigEngine.sh database-transfer -DWasPassword=password`

**Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command: `./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password`

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

7. From the command line processor, reset the check pending status on the WebSphere Portal table spaces listed here.

CHECK DATA is a DB2 batch utility that is invoked through JCL or through the DB2 utility panels. Type the following utility commands to check pending status:

```bash
check data tablespace releasenameonzos.TS320A
check data tablespace releasenameonzos.TS300A
check data tablespace releasenameonzos.TS2110A
check data tablespace releasenameonzos.TS830A
check data tablespace communitynameonzos.TS8000B
check data tablespace communitynameonzos.TS8011B
check data tablespace communitynameonzos.TS280B
check data tablespace customizationnameonzos.TS2110C
check data tablespace jcrdbnameonzos.ICMSFQ04
check data tablespace jcrdbnameonzos.TS2110D
```

where `releasenameonzos`, `communitynameonzos`, and `customizationnameonzos` are the names of your WebSphere Portal databases, and `jcrdbnameonzos` is the name of your JCR database. Refer to the Utility Guide and Reference for DB2 for z/OS for additional details.

8. Run the `RUNSTATS` utility as shown in the following example:

```bash
LISTDEF JCRDBZOS INCLUDE TABLESPACE JCRDBZOS.*  BASE
RUNSTATS TABLESPACE LIST JCRDBZOS INDEX(ALL) KEYCARD TABLE(ALL)
LISTDEF JCRDB01 INCLUDE TABLESPACE JCRDB01.*  BASE
RUNSTATS TABLESPACE LIST JCRDB01 INDEX(ALL) KEYCARD TABLE(ALL)
...
```

9. Start the WebSphere Application Server.

10. Verify that the WebSphere Portal application server is running by opening the following URL in a browser: `http://hostname.companyname.com:port_number/wps/portal`, where `hostname.companyname.com` is the fully qualified host name of the machine where WebSphere Portal is running and `port_number` is the transport port that is created by WebSphere Application Server.

If you have performed database transfer in a clustered environment, you must explicitly copy the `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm/icm.properties` file from the primary node on which the `database-transfer` task was processed to all secondary nodes. This ensures that the secondary nodes have the new JCR database values specified in the `icm.properties` file.

1. Stop the portal server on the secondary node.
2. Copy the `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm/icm.properties` file from the primary node and replace the `icm.properties` file on the secondary node with the new file from the primary node.
3. Start the portal server on the secondary node.
Verifying database connections

After WebSphere Portal is configured to work with your database, test the database connection to ensure that it operates correctly.

- **Prerequisites**
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases
  - Creating the Java Content Repository database
  - Modifying database properties
  - Configuring WebSphere Portal to use DB2 for z/OS

You can verify the connection from a browser or from a command line. To verify that WebSphere Portal is running from a browser, open the portal in a Web browser: http://hostname.yourco.com:port_number/wps/portal, where hostname.yourco.com is the fully qualified host name of the machine where WebSphere Portal is running and port_number is the transport port that is created by IBM® WebSphere® Application Server.

There may be an error if any of the following conditions appear.

- When trying to access the portal you get a 503 error.
- If you had any locale problems with your database, you could see invalid characters, such as ?????, after logging in. This may happen if the character set of the database is not UTF-8 compliant.
- If something went wrong with the data that was transferred, you may not be able to login. WebSphere Portal will indicate you entered an invalid user ID and password even though you know it is valid.

Verify the connection from a command line by completing the following steps:

1. Start a 5250 session on the local machine where WebSphere Portal is installed.
2. For WebSphere Portal on WebSphere Application Server (UserData path), enter the following on the command line: cd wp_profile_root/ConfigEngine.
3. Enter the following command: ConfigEngine.sh validate-database-connection -DTransferDomainList="release,community,customization,jcr,feedback,likeminds" -DWasPassword=*

For security reasons, you should not leave passwords in the wkplc_comp.properties file. Edit the file prior to running a configuration task and insert the passwords that are needed for that task. After the task has run, delete all passwords from the file.

Alternatively, you can specify the password on the command line rather than update the wkplc_comp.properties file. For example: ConfigEngine.sh -DPortalAdminPwd=*

When installing WebSphere Portal, the passwords in the wkplc_comp.properties file are automatically removed after configuration.

**Parent topic:** Preparing DB2 for z/OS for a stand-alone production server

**Previous topic:** Configuring WebSphere Portal to use DB2 for z/OS
Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Setting up the Oracle database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

1. Installing Oracle
   View information on how to install Oracle for use with WebSphere Portal.

2. Creating databases
   View some important considerations before setting up Oracle databases to work with WebSphere Portal.

3. Creating users
   View the steps to set up users for Oracle to work with WebSphere Portal.

4. Modifying database properties
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

6. Solaris stand-alone server: Creating JCR table spaces
   This topic provides manual instructions for creating JCR table spaces for Oracle.

7. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

8. Configure WebSphere Portal to use Oracle
   This section provides information on how to manually transfer data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configuring databases
Installing Oracle

View information on how to install Oracle for use with WebSphere Portal.

You can use Oracle as the database software. You must install Oracle separately and before performing a database transfer.

Refer to Oracle product documentation for installation instructions.

Parent topic: Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Next topic: Creating databases
Creating databases

View some important considerations before setting up Oracle databases to work with WebSphere Portal.

For information about creating databases, refer to the Oracle product documentation. For information on the recommended database architecture and the databases you will need to create, see the Planning for Oracle topic. Be sure that all databases to be used with WebSphere Portal are created as UNICODE character set databases.

If you are using remote Oracle 9 or 10 databases, you must also copy the ojdbc14.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM® WebSphere® Application Server Version 7.0, you must also copy the ojdbc6.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM WebSphere Application Server Version 6.1, you must also copy the ojdbc5.jar file from the remote Oracle server to the WebSphere Portal machine. The typical location is the oracle_home/sqldeveloper/jdbc/lib directory. Record the copy location on your local machine for future reference.

When creating Oracle databases for use with WebSphere Portal, you should consider the following information:

- All databases must be created manually before configuring WebSphere Portal.
- All databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle server will be remote or local to the WebSphere Portal installation.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle database in order for WebSphere Portal to communicate with the Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes:

  - `db_block_size = 8192 bytes`
  - `db_cache_size = 307200 bytes`
  - `db_files = 1024 files`
  - `log_buffer = 65536 bytes`
  - `open_cursors = 1500 cursors`
  - `pga_aggregate_target = 204800 bytes`
  - `pre_page_sga = true`
  - `processes = 300 processes`
  - `shared_pool_size = 204800 bytes`

  **Note:** If you are using IBM Java Content Repository, the `open_cursors` value may need to be increased based on the table count in the Java Content Repository schema.

- Raise the number of parallel servers as appropriate. For example, if you have more than 875 parallel servers, you should set the `parallel_max_servers` to 1200.
- The Oracle parameter CURSOR_SHARING allows similar SQL Statements to be shared when possible, which prevents parsing and establishing a new execution plan. The execution plan is used by Oracle to gather the data needed to satisfy a request. There are two options for CURSOR_SHARING, which are as follows:
- **FORCE**
  - When you select this option, Oracle uses the same execution plan for all SQLs that are similar in value even if the values are different. When you use this option, the execution plan may not provide optimum performance. For example, similar SQLs with different values may behave differently when executed running the same plan.
- **EXACT**
When you select this option, Oracle only shares the same execution plan for SQLs that are identical and use the same values. This option removes the risk of a SQL statement being executed when optimum performance conditions do not exist.

WebSphere Portal supports both options. Regardless of the option selected, portlet applications should not be affected. Contact your database administrator for further assistance on these options.

- **Prerequisites**
  - Installing Oracle

Parent topic: Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server
Previous topic: Installing Oracle
Next topic: Creating users
Creating users

View the steps to set up users for Oracle to work with WebSphere Portal.

Before you begin:

- You should have installed Oracle.

- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

**Prerequisites**

- Installing Oracle
- Creating databases

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Start the Oracle SQL command line processor ‘SQL*Plus’ by entering `sqlplus /nolog` on the operating system command prompt.

2. Log in to the Oracle database by executing command `connect user/password@database name`. For example: `system/manager@wpsdb` will log the administrative user `system` with a password of `manager` into the `wpsdb` database.

3. Create the WebSphere Portal user `dbdomain.DbUser`, where `dbdomain` is replaced by release, community, or customization.

   A. Log in to the database in which you want to create the new users.

   B. Create the users that you need in the appropriate databases by entering the following commands in the SQL*Plus tool, where `user_tablespace` is the default tablespace that is identified by the database administrator to store user objects and `temp_tablespace` is identified to store temporary objects. All names must be in lower case:

   ```sql
   SQL> create user releaseusr identified by password
   default tablespace user_tablespace
   temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO releaseusr;
   SQL> create user communityusr identified by password
   default tablespace user_tablespace temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO communityusr;
   SQL> create user customizationusr identified by password
   default tablespace user_tablespace
   temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO customizationusr;
   SQL> create user jcr identified by password
   default tablespace user_tablespace
   temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO jcr
   ```

4. Connect to the Feedback database:

   A. Enter the following command: `SQL> connect`
B. Enter `user-name: username/password@dbname` where `username` is an existing administrative user in the database. For example, `system/manager@fdbkdb` will log the administrative user system with a password of manager into the fdbkdb database.

C. Create the Feedback user:

```sql
SQL> create user feedback identified by password
    
default tablespace user_tablespace
    
temporary tablespace temp_tablespace;
```

GRANT UNLIMITED TABLESPACE TO feedback

D. Log out of the command line tool using the command `SQL> exit`.

5. Connect to the LikeMinds database:

A. Enter the following command:

```sql
SQL> connect
```

B. Enter `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example, `system/manager@lmdb` will log the administrative user system with a password of manager into the lmdb database.

C. Create the LikeMinds user:

```sql
SQL> create user lmdbusr identified by password
    
default tablespace user_tablespace
    
temporary tablespace temp_tablespace;
```

GRANT UNLIMITED TABLESPACE TO lmdbusr

D. Log out of the command line tool using the command `SQL> exit`.

6. Perform the following steps to verify that you can connect to the database with the user ID and password that you just created:

A. Enter the `SQL> connect` command to connect to the content database.

B. Enter `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example, `system/manager@jcrdb` will log the administrative user system with a password of manager into the Oracle database.
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Installing Oracle
  - Creating databases
  - Creating users

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**

- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:

  - release.DbName = release
  - jcr.DbName = jcrdb
  - feedback.DbName = fdbkdb
  - likeminds.DbName = lmdb
  - community.DbName = commdb
  - customization.DbName = custdb

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds

- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

1. Locate the following files and create a backup copy of each before changing any values:

   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.

   A. For dbdomainDbType, type oracle.

   B. For dbdomainDbName, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the dbdomainDbUrl property.

   C. For dbdomainDbSchema, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The dbdomainDbName should be the same value used for the dbdomainDbSchema.

   **Restriction:** The value for dbdomainDbSchema must equal the value for dbdomainDbUser.

   D. For dbdomainData SourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:

      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback

   E. For dbdomainDbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:**

      - The database element of this value should match the value of DbName.

      - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following:

        jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME. **When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.**
F. For \texttt{dbdomain}.DbUser, type the user ID for the database administrator. \textbf{Restriction}: The value for \texttt{dbdomain} .DbUser must equal the value for \texttt{dbdomain}.DbSchema.

G. For \texttt{dbdomain}.DbPassword, type the password for the database administrator.

H. For \texttt{dbdomain}.DBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.

I. For \texttt{dbdomain}.DBA.DbPassword, type the database administrator password for privileged access operations during creation of the database.

J. For \texttt{dbdomain}.DbHome, type the root location for the database. \textbf{Note}: This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in \texttt{wkplc_comp.properties}. \textbf{Important}: The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For \texttt{source.domain}.DbType, type of the database you are currently configured to use. The value for \texttt{source.domain} .DbType is Derby by default.

B. For \texttt{source.domain}.DbName, type the name of the database domain you are currently using.

C. For \texttt{source.domain}.DbSchema, type current schema identifier for objects within the database for this domain.

D. For \texttt{source.domain}.DataSourceName, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For \texttt{source.domain}.DbUrl, type the url currently used to access your database.

F. For \texttt{source.domain}.DbUser, type the name of the user accessing this database.

G. For \texttt{source.domain}.DbPassword, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file \texttt{wkplc_dbtype.properties}.

A. For oracle.DbDriver, type the name of the class that SqlProcessor uses to import SQL files.

B. For oracle.DbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

C. For oracle.JdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file \texttt{wkplc.properties}.

A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

\textbf{Parent topic:} Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server \\
\textbf{Previous topic:} Creating users \\
\textbf{Next topic:} Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

**Prerequisites**
- Installing Oracle
- Creating databases
- Creating users
- Modifying database properties

1. On the database server, make sure that the folder `db_1` contains the subfolders `your_oracle_instance/data` and `your_oracle_instance/index`. If this folder hierarchy does not exist, create it manually before you run the `setup-database` task. The `setup-database` task requires these folders to create database users. If these folders do not exist, the `setup-database` task will fail.

2. Change to the directory `wp_profile_root/ConfigEngine`

3. To create the database users, type the following command:

   ```
   ./ConfigEngine.sh setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:

   `':<WP_root>ase\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql'`.

---

**Parent topic:** Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Modifying database properties

**Next topic:** Solaris stand-alone server: Creating JCR table spaces

**Related tasks**
Manually creating users and granting privileges for Oracle
Solaris stand-alone server: Creating JCR table spaces

This topic provides manual instructions for creating JCR table spaces for Oracle.

- Prerequisites
  - Installing Oracle
  - Creating databases
  - Creating users
  - Modifying database properties
  - Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.
2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:
   - `&jcrdb` is the name of the database you created to store user data.
   - `&dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.

   Important: You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

```
cREATE TABLESPACE ICMLFQ32 DATAFILE 
&dbpath./&jcrdb./data/ICMLFQ32_01.dbf SIZE 300M REUSE AUTOEXTEND ON
NEXT 10M MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
```

```
cREATE TABLESPACE ICMLNF32 DATAFILE 
&dbpath./&jcrdb./data/ICMLNF32_01.dbf SIZE 25M REUSE AUTOEXTEND ON
NEXT 10M MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
```

```
cREATE TABLESPACE ICMMVFQ04 DATAFILE 
&dbpath./&jcrdb./data/ICMMVFQ04_01.dbf SIZE 150M REUSE AUTOEXTEND ON
NEXT 10M MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
```

```
cREATE TABLESPACE ICMLSNDX DATAFILE 
&dbpath./&jcrdb./index/ICMLSNDX_01.dbf SIZE 10M REUSE AUTOEXTEND ON
NEXT 10M MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
```

A. Set the size, autoextend, and maxsize values according to your environment. For example, you may want to change the maxsize to a set value rather than UNLIMITED.

B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.

C. Refer to the Oracle command reference for more information about using the `create tablespace` command.

Parent topic: Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Setting up databases

Next topic: Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

**- Prerequisites**

- Installing Oracle
- Creating databases
- Creating users
- Modifying database properties
- Setting up databases
- Solaris stand-alone server: Creating JCR table spaces

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.index_name.indexspace
   
   For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds

3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword TABLESPACE and a space. For example: community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE

   Repeat this step for each domain that you are transferring.

4. Save and close dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,

   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

Parent topic: Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server
Previous topic: Solaris stand-alone server: Creating JCR table spaces
Next topic: Configure WebSphere Portal to use Oracle
Configure WebSphere Portal to use Oracle

This section provides information on how to manually transfer data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- Prerequisites
  - Installing Oracle
  - Creating databases
  - Creating users
  - Modifying database properties
  - Setting up databases
  - Solaris stand-alone server: Creating JCR table spaces

Tips:
- If you are transferring from Oracle, the `open_cursors` setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

Steps for transferring data to another supported database

1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.
2. Enter the following commands to validate configuration properties:
   ```bash
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ```
3. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.
4. Stop both WebSphere Application Server and the WebSphere Portal server:
   ```bash
   ./stopServer.sh server1 -username admin_userid -password admin_password
   ```
5. Transfer the database. **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory `wp_profile_root/ConfigEngine`.

   B. Enter the following command:
   ```bash
   ./ConfigEngine.sh database-transfer -DWasPassword=password
   ```
   
   **Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:
   ```bash
   ./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
   ```

   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

   If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example:
   ```sql
   SQL> execute dbms_stats.gather_schema_stats(ownname=> 'jcr', cascade=> TRUE);
   ```

7. Change to the directory `wp_profile_root/bin`.

8. Enter the following command to start the WebSphere Portal server:
   ```bash
   ./startServer.sh WebSphere_Portal
   ```

---

**Parent topic:** Setting up a remote Oracle database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Assigning custom table spaces
Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Setting up the Oracle RAC database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

1. **Installing Oracle RAC**
   View information on how to install Oracle RAC for use with WebSphere Portal.

2. **Create users**
   View the steps to set up users for Oracle RAC to work with WebSphere Portal.

3. **Creating databases**
   View some important considerations before setting up Oracle RAC databases to work with WebSphere Portal.

4. **Modifying database properties**
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. **Setting up databases**
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

6. **Solaris stand-alone server: Creating JCR table spaces**
   This topic provides manual instructions for creating JCR table spaces for Oracle.

7. **Optional: Assigning custom table spaces**
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

8. **Configuring WebSphere Portal to use Oracle RAC**
   This section provides information on how to manually transfer data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

**Parent topic:** Configuring databases
Installing Oracle RAC

View information on how to install Oracle RAC for use with WebSphere Portal.

Before you begin:
- You should have completed reviewing the Planning for Oracle RAC section.
- Ensure the Oracle CRS (Cluster Ready Service) and Oracle RAC databases have been installed and configured on the primary and secondary nodes.
- Run the following commands on both RAC nodes to start global services daemon_(GSD), oracle listeners, and agents.
  - $ gsdctl start
  - $ lsnrctl start
  - $ agentctl start

You can use Oracle RAC as the database software for WebSphere Portal. You must install Oracle RAC separately and before performing WebSphere Portal database transfer.

Refer to Oracle product documentation for installation instructions.

**Parent topic:** Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Next topic:** Create users
Create users

View the steps to set up users for Oracle RAC to work with WebSphere Portal.

Before you begin:
- You should have completed installing Oracle RAC.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

**Prerequisites**
- Installing Oracle RAC

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Create the users in the WebSphere Portal databases
   A. Log in to the database in which you want to create the new users.
   B. Create the users that you need in the appropriate databases by entering the following command in the SQL*Plus tool, where `user_tablespace` is the default tablespace that is identified by the database administrator to store user objects and `temp_tablespace` is identified to store temporary objects: **Tip:** Balance the storage of user objects among tablespaces to prevent running out of space with overuse of `user_tablespace`. Also consider increasing the size of `user_tablespace` when handling a high volume of users.

   ```sql
   SQL> create user username identified by password
       default tablespace user_tablespace
       temporary tablespace temp_tablespace;
   ```

2. Log in by entering the command `$ sqlplus` in SQL*Plus:
3. Enter the command `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example: `system/manager@wpsdb` will log the administrative user system with a password of manager into the `wpsdb` database.
4. Create the WebSphere Portal user `dbdomain.dbUser`, where `dbdomain` is replaced by release, community, customization, or JCR. All names must be in lower case. The following code uses the default user tablespace and the default temporary tablespace. Explicitly specify these tablespaces if you do not want to use the defaults.

   ```sql
   SQL> create user releaseusr identified by password;
   SQL> create user communityusr identified by password;
   SQL> create user customizationusr identified by password;
   SQL> create user jcr identified by password;
   ```

5. Connect to the content database by entering the command `SQL> connect`.
6. Enter the following, where `username` is an existing administrative user in the database: `user-name: username/password@dbname` For example, `system/manager@jcrdb` will log the administrative user system with a password of manager into the Oracle RAC database.
7. Create the Oracle RAC users by entering the following: When creating the `jcr` user, grant all necessary privileges. If you do not grant privileges, you will receive the error `ICMADMIN lacks CREATE SESSION Privilege logon denied`
when you try to connect with the jcr user:

SQL> create user jcr identified by password default tablespace users temporary tablespace temp;

8. Connect to the Feedback database:
   A. Enter the following command: SQL> connect
   B. Enter `user-name: username/password@dbname` where `username` is an existing administrative user in the database. For example, `system/manager@fdbkdb` will log the administrative user system with a password of manager into the `fdbkdb` database.

9. Create the Feedback user:
   SQL> create user feedback identified by password default tablespace users temporary tablespace temp;

10. Connect to the LikeMinds database:
    SQL> connect

11. Enter `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example, `system/manager@lmdb` will log the administrative user system with a password of manager into the `lmdb` database.

12. Create the LikeMinds user:
    SQL> create user lmdbusr identified by password default tablespace users temporary tablespace temp;

13. Log out of the command line tool using the command SQL> exit.

Parent topic: Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Installing Oracle RAC

Next topic: Creating databases
Creating databases

View some important considerations before setting up Oracle RAC databases to work with WebSphere Portal.

When creating Oracle RAC databases for use with WebSphere Portal, you should consider the following information:

- The Oracle RAC databases must be created manually before configuring WebSphere Portal.
- All databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle RAC server will be remote or local to the WebSphere Portal installation.
- If using an earlier version of Oracle RAC (9i or earlier), ensure that Oracle RAC JVM is also installed.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle RAC database in order for WebSphere Portal to communicate with the IBM Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes are:

  - `db_block_size = 8192` bytes
  - `db_cache_size = 314,572,800` bytes
  - `db_files = 1024` files
  - `log_buffer = 65536` bytes
  - `open_cursors = 1500` cursors
  - `pga_aggregate_target = 209,715,200` bytes
  - `pre_page_sga = true`
  - `processes = 300` processes
  - `shared_pool_size = 209,715,200` bytes

  **Note:** If you are using IBM Java Content Repository, the `open_cursors` value may need to be increased based on the table count in the IBM Java Content Repository schema.

**Prerequisites**

- Installing Oracle RAC
- Create users

Refer to the following instructions to create tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.

2. Create the tablespaces. **Important:** You must use the same tablespace names listed in the commands. The tablespace names cannot be customized or modified.

   A. Find and edit the SQL script `jcr_ora_tablespaces.sql` in the directory `wp_profile_root/ConfigEngine/work/db/oracle`.

   B. In the define section, replace the following variables with the values from your environment:

      - `jcrdb`
          - Name of the database you created to store user data.

      - `logfile`
          - Location to store the log file.

      - `dbpath`
          - Directory where you created the database.

   C. Set the size, autoextend, and maxsize values according to your environment. For example, you may want to change the maxsize to a set value rather than UNLIMITED. Consult your Database Administrator for more info.
D. Execute the following SQL script:

Notes:
- The DBA or a user with sufficient credentials (CREATE tablespace) must execute the script.
- The script will prompt for the database username and password.

```sql
# sqlplus
SQL > @jcr_ora_tablespaces.sql
```

Parent topic: Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server
Previous topic: Create users
Next topic: Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Installing Oracle RAC
  - Create users
  - Creating databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**

- The WebSphere Portal database can be used to hold information for applications such as Feedback and Like Minds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:
  - release.DbName = release
  - jcr.DbName = jcrdb
  - feedback.DbName = fdbkdb
  - likeminds.DbName = lmdb
  - community.DbName = commdb
  - customization.DbName = custdb

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

1. Locate the following files and create a backup copy of each before changing any values:
   - `wp_profile_root/ConfigEngine/properties/wkplc.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_comp.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties`

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.

A. For `dbdomain.DbType`, type `oracle`.

B. For `dbdomain.DbName`, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the `dbdomain.DbUrl` property.

C. For `dbdomain.DbSchema`, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The dbdomain.DbName should be the same value used for the dbdomain.DbSchema

   **Restriction:** The value for `dbdomain.DbSchema` must equal the value for `dbdomain.DbUser`.

D. For `dbdomain.DataSourceName`, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
   - `releaseDS`
   - `communityDS`
   - `customizationDS`
   - `jcrDS`
   - `lmdbDS`
   - `feedback`

E. For `dbdomain.DbUrl`, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:**

   - The database element of this value should match the value of DbName.

   - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following:

     `jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME`. **When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.**
F. For `dbdomain.DbUser`, type the user ID for the database administrator. **Restriction:** The value for `dbdomain.DbUser` must equal the value for `dbdomain.DbSchema`.

G. For `dbdomain.DbPassword`, type the password for the database administrator.

H. For `dbdomain.DBA.DbUser`, type the database administrator user ID for privileged access operations during creation of the database.

I. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.

J. For `dbdomain.DbHome`, type the root location for the database. **Note:** This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

   A. For `source.domainDbType`, type of the database you are currently configured to use. The value for `source.domainDbType` is Derby by default.

   B. For `source.domain.DbName`, type the name of the database domain you are currently using.

   C. For `source.domain.DbSchema`, type current schema identifier for objects within the database for this domain.

   D. For `source.domain.DataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

   E. For `source.domain.DbUrl`, type the url currently used to access your database.

   F. For `source.domain.DbUser`, type the name of the user accessing this database.

   G. For `source.domain.DbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

   A. For `oracle.DbDriver`, type the name of the class that `SqlProcessor` uses to import SQL files.

   B. For `oracle.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For `oracle.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Creating databases

**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

Before you begin, ensure that the following prerequisites are met:
- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

- Prerequisites
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties

1. On the database server, make sure that the folder `db_1` contains the subfolders `your_oracle_instance/data` and `your_oracle_instance/index`. If this folder hierarchy does not exist, create it manually before you run the `setup-database` task. The `setup-database` task requires these folders to create database users. If these folders do not exist, the `setup-database` task will fail.

2. Change to the directory `wp_profile_root/ConfigEngine`

3. To create the database users, type the following command:

   ```bash
   ./ConfigEngine.sh setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:
   ```bash
   <WP_root>/base/wp.db.impl/config/templates/setupdb/sqlserver2005\<database domain>\createUsers.sql
   ```

Parent topic: Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Modifying database properties

Next topic: Solaris stand-alone server: Creating JCR table spaces

Related tasks
- Manually creating users and granting privileges for Oracle
Solaris stand-alone server: Creating JCR tablespaces

This topic provides manual instructions for creating JCR table spaces for Oracle.

- **Prerequisites**
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties
  - Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.
2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:
   - `&jcrdb` is the name of the database you created to store user data.
   - `&dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.

   Ensure that the `.` is included in the variables when you substitute the values of your environment with these variables.

   Important: You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

   ```sql
   create tablespace ICMLFQ32 datafile '"dbpath./&jcrdb./data/ICMLFQ32_01.dbf" size 300M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLNF32 datafile '"dbpath./&jcrdb./data/ICMLNF32_01.dbf" size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMVFQ04 datafile '"dbpath./&jcrdb./data/ICMVFQ04_01.dbf" size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMSFQ04 datafile '"dbpath./&jcrdb./data/ICMSFQ04_01.dbf" size 150M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLSNDX datafile '"dbpath./&jcrdb./index/ICMLSNDX_01.dbf" size 10M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   ```

   A. Set the `size`, `autoextend`, and `maxsize` values according to your environment. For example, you may want to change the `maxsize` to a set value rather than `UNLIMITED`.
   
   B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.
   
   C. Refer to the Oracle command reference for more information about using the `create tablespace` command.

---

**Parent topic:** Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server  
**Previous topic:** Setting up databases  
**Next topic:** Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:

- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the wp_profile_root/PortalServer/config/tables Bossages/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

- Prerequisites
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties
  - Setting up databases
  - Solaris stand-alone server: Creating JCR table spaces

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file wp_profile_root/PortalServer/config/tables Bossages/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.index_name.indexspace
   For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword TABLESPACE and a space. For example: community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE
   Repeat this step for each domain that you are transferring.
4. Save and close dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

i5/OS: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

Parent topic: Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

Previous topic: Solaris stand-alone server: Creating JCR table spaces

Next topic: Configuring WebSphere Portal to use Oracle RAC
Configuring WebSphere Portal to use Oracle RAC

This section provides information on how to manually transfer data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- **Prerequisites**
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties
  - Setting up databases
  - Solaris stand-alone server: Creating JCR table spaces

**Tips:**

- If you are transferring from Oracle, the `open_cursors` setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

- The WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following: `jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME`. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.

- Manually specify an RAC datasource URL in the WebSphere Application Server console. The URL should look like the following:

  ```sql
  jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT=1521))
  ) (ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICENAME))
  ```

- When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

**Steps for transferring data to another supported database**
1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.
2. Enter the following commands to validate configuration properties:
   ```
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```
3. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.
4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td><code>./stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td><code>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

5. Transfer the database:
   - Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.
   - A. Change to the directory `wp_profile_root/ConfigEngine`.
   - B. Enter the following command:
     ```
     .\ConfigEngine.sh database-transfer -DWasPassword=password
     ```
     **Note:** To select specific database domains to transfer, modify the `DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:
     ```
     .\ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
     ```
   - C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.
   - If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example:
   ```
   SQL> execute dbms_stats.gather_schema_stats(ownname='jcr', cascade=> TRUE);
   ```

7. Specify the JDBC URL to connect to the cluster:
   - A. Login to the WebSphere Application Server Administrator Console
   - B. Navigate to Resources > JDBC Providers
   - C. If there is a value in the Node field, remove it and click Apply.
   - D. For each Oracle JDBC provider, repeat the following steps:
     1. Click the provider name.
     2. Select Data Sources.
     3. Click the name of the data source.
     4. Under the Oracle data source properties, modify the URL attribute according to Oracle RAC configuration:
        ```
        jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICENAME)))
        ```
     5. Save your changes

8. Change to the directory `wp_profile_root/bin`.
9. Enter the following command to start the WebSphere Portal server:
   ```
   .\startServer.sh WebSphere_Portal
   ```

**Parent topic:** Setting up a remote Oracle RAC database on AIX, HP-UX, Linux, and Solaris for a stand-alone production server

**Previous topic:** Assigning custom table spaces
Setting up a remote SQL Server database on Windows for a stand-alone production server

To setup a remote Microsoft SQL Server Enterprise Edition database, create user IDs and databases on a remote server. Tasks are provided to assist with creating the user IDs and the databases. Before you can use the tasks, you must modify properties files.

1. **Installing SQL Server**
   View the steps to install SQL Server for use with WebSphere Portal.

2. **Modifying database properties**
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

3. **Setting up databases**
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

   The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

5. **Configuring WebSphere Portal to use SQL Server 2005**
   View information on manually transferring data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

**Parent topic:** Configuring databases
Installing SQL Server

View the steps to install SQL Server for use with WebSphere Portal. Before you begin this task, complete the following prerequisites:

- You should have completed reviewing the Planning for SQL Server topic.
- You must install SQL Server separately from WebSphere Portal.
- WebSphere Portal requires either the DataDirect Connect for JDBC 3.7 JDBC driver or the Microsoft SQL Server JDBC Driver.

Before installing SQL Server, you can obtain the DataDirect Driver from DataDirect Technologies. The JDBC driver consists of two parts: the JDBC driver libraries and JTA libraries. The JDBC driver libraries must be put on the WebSphere Portal system, while the JTA libraries are intended for SQL Server.

You can obtain the Microsoft SQL Server JDBC Driver from Microsoft. See the Microsoft SQL Server product documentation for installation details.

The following section provides instructions for installing SQL Server for use with IBM® WebSphere® Application Server and WebSphere Portal. These steps are the same for both the DataDirect and Microsoft drivers unless noted.

1. Install SQL Server and all required patches.
2. Select the authentication mode for this installation:
   - Windows Authentication Mode
   - Mixed Mode (Windows Authentication and SQL Server Authentication)

   Either authentication mode allows the user who is specified as the database administrator in the wkplc_comp.properties file to log in to the SQL server. Mixed Mode authentication allows either a Windows user or a SQL server user, or both, to log in to the SQL server.

3. In the SQL Server Setup panel, Components to Install, select the following components, which are required services for WebSphere Portal:
   - SQL Server Database Services
     - Integration Services

   The option Integration Services creates the directory DTS/Binn, where you will need to copy files when you install Microsoft SQL Server JDBC drivers and enable XA connections.

4. Complete the installation using SQL Server documentation as a guide.
5. Enable TCP/IP connectivity in the SQL Server Configuration Manager.
6. Install the JDBC driver using one of these methods:
   - Installing DataDirect Connect for JDBC drivers on UNIX
   - Installing DataDirect Connect for JDBC drivers on Windows
   - Installing Microsoft SQL Server JDBC drivers and enabling XA connections

7. Network the connections for MSDTC in the Windows Component Services as described in Microsoft Help and Support.
8. Start SQL Server.

**Installing DataDirect Connect for JDBC drivers on UNIX**

1. Purchase and download DataDirect Connect for JDBC and save file 360connectjdbc.jar in a temporary work directory.
To create the required files, run the following command from the directory that contains 360connectjdbc.jar:

```
jar -xvf 360connectjdbc.jar
```

3. Run `./Installer.sh` in the same directory.

4. When installation is complete, change directory to `/installation/lib`, where `installation` is the directory in which you installed the DataDirect Connect for JDBC drivers.

5. Run the following command to change the permissions on the JAR files in the `installation/lib` directory:

   ```
   chmod 777 *.jar
   ```

6. Change the ownership and group of these files by running the following commands in the `installation/lib` directory:

   ```
   chgrp system_grp *.jar
   chown root *.jar
   ```

   Where `system_grp` is the system group as labeled by your operating system.

### Installing DataDirect Connect for JDBC drivers on Windows

1. Purchase and download DataDirect Connect for JDBC and save file `360connectjdbc.jar` in a temporary work directory.

2. To create the required files, run the following command from the directory that contains `360connectjdbc.jar`:

   ```
   jar -xvf 360connectjdbc.jar
   ```

3. Run `Installer.bat` in the same directory.

### Installing Microsoft SQL Server JDBC drivers and enabling XA connections

1. Download and install the Microsoft SQL Server JDBC driver.

2. Copy file `sqljdbc_xa.dll` from the `xa` subdirectory to the `C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqljdbc_xa.dll` directory of the SQL Server installation.

3. Start the database server.

4. Ensure that the Distributed Transaction Coordinator has been started. The status can be verified in the list of services in the Computer Management console.

5. Start the Microsoft SQL Server Management Studio and connect to the local database engine as the system administrator, `sa`.

6. Select `File > Open > File` and select `xa_install.sql` from the subdirectory of the downloaded and extracted JDBC driver.

7. Execute the script by selecting `Query > Execute`. Note: Any warnings that appear in the messages section of the application window that say that stored procedures cannot be found can be safely ignored.


   A. Open the Windows Registry Editor (regedit) and navigate to the element

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSDTC\XADLL
   ```

   B. From the menu bar, select `Edit > New > String Value` to create a new parameter named `sqljdbc_xa.dll` in that element.

   C. Change the value of the new parameter to the location of the `sqljdbc_xa.dll` file copied in Step 2 above, for example:

   ```
   C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqljdbc_xa.dll
   ```

### Enable XA Transactions in Windows Component Services

1. From your Windows desktop, follow these steps:

   2. Click `Start > Settings > Administrative Tools > Component Services.`
3. Expand the tree view to locate the computer where you want to turn on support for XA transactions (for example, My Computer).

4. Display the context menu for the computer name and click Properties.

5. Click Options and tune the Transaction Timeout that suits your environment. (The recommended minimum is 180 seconds).

6. Click MSDTC and click Security Configuration.

7. Under Security Settings, select XA Transactions to enable this support.

8. Click OK to save your changes.

**Note:** The installation documentation for JDBC XA connectivity refers to two known problems, see the Microsoft support site for more information:

- KB318818: Performance slows down when you use XA Transactions with SQL Server.

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server

**Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Installing SQL Server

  The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

  **Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
  - WebSphere Portal 6.1.5 wkplc.properties file reference
  - WebSphere Portal 6.1.5 wkplc_comp.properties file reference
  - WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

- **Working with properties files:**

  - The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:
    - `release.DbName=release`
    - `jcr.DbName=jcrdb`
    - `feedback.DbName=fdbkdb`
    - `likeminds.DbName=lmdb`
    - `community.DbName=commdb`
    - `customization.DbName=custdb`

  - If you are using a remote database, enter the values for the remote server.

  - Use a forward slash (/) instead of a backslash (/).

  - There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.

  - The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.

  - Depending on which database domain has to be configured, replace dbdomain with:
    - `release`
    - `customization`
    - `community`
    - `jcr`
    - `feedback`
    - `likeminds`

  - The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
    - `dbdomain.DbType`
    - `dbdomain.DbName`
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If \textit{DbUser}, \textit{DbUrl}, and \textit{DbPassword} are not the same across domains, the value for \textit{DataSourceName} must differ from the \textit{DataSourceName} of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   - \texttt{wp\_profile\_root/ConfigEngine/properties/wkplc.properties}
   - \texttt{wp\_profile\_root/ConfigEngine/properties/wkplc\_comp.properties}
   - \texttt{wp\_profile\_root/ConfigEngine/properties/wkplc\_dbtype.properties}

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In \texttt{wkplc\_comp.properties}, most properties are repeated for each domain.

2. Use a text editor to open the properties file \texttt{wkplc\_comp.properties} and modify the values to correspond to your environment.
   
   A. For \texttt{dbdomainDbType}, type \texttt{sqlserver2005}.
   
   B. For \texttt{dbdomainDbName}, type the name of the WebSphere Portal domain database. \textbf{Note:} This value is also the database element in the \texttt{dbdomainDbUrl} property.
   
   C. For \texttt{dbdomainDbSchema}, type the schema name of the database domain. \textbf{Note:} Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   
   D. For \texttt{dbdomainDataSourceName}, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - \texttt{releaseDS}
      - \texttt{communityDS}
      - \texttt{customizationDS}
      - \texttt{jcrDS}
      - \texttt{lmdbDS}
      - \texttt{feedback}
   
   E. For \texttt{dbdomainDbUrl}, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. \textbf{Note:} The database element of this value should match the value of \texttt{DbName}.
   
   F. For \texttt{dbdomainDbUser}, type the user ID for the database administrator.
   
   G. For \texttt{dbdomainDbPassword}, type the password for the database administrator.
   
   H. For \texttt{dbdomainDBA\_DbUser}, type the database administrator user ID for privileged access operations during creation of the database.
   
   I. For \texttt{dbdomainDBA\_DbPassword}, type the database administrator password for privileged access operations during creation of the database.
   
   J. For \texttt{dbdomainDbHome}, type the root location for the database. \textbf{Note:}
      - This value is the location to store the database files locally.
      - This path must use two backslashes (\textbackslash) instead of a forward slash (/).
   
   K. For \texttt{dbdomainAdminUrl}, type the sqlserver URL without a database attached. This value is used to connect to the server for database administration operations.
   
   L. For \texttt{dbdomainDbHostName}, type the hostname of the database.
3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For `source.domain.DbType`, type of the database you are currently configured to use. The value for `source.domain.DbType` is Derby by default.
B. For `source.domain.DbName`, type the name of the database domain you are currently using.
C. For `source.domain.DbSchema`, type current schema identifier for objects within the database for this domain.
D. For `source.domain.DataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.
E. For `source.domain.DbUrl`, type the url currently used to access your database.
F. For `source.domain.DbUser`, type the name of the user accessing this database.
G. For `source.domain.DbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.
   A. For `sqlserver2005.DbDriver`, type the name of the class that SqlProcessor uses to import SQL files.
   B. For `sqlserver2005.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.
   C. For `sqlserver2005.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.
   D. For `sqlserver2005.DbConnectionPoolDataSource`, type the name of the implementation class of the connection pool data source.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.
   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server

**Previous topic:** Installing SQL Server

**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

Before you begin, ensure that the following prerequisites are met:

- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

**Prerequisites**

- Installing SQL Server
- Modifying database properties

1. Change to the directory `wp_profile_root/ConfigEngine`
2. To create the databases, type the following command:
   ```cmd
   ConfigEngine.bat create-database -DWasPassword=password
   ```
3. To create the database users, type the following command:
   ```cmd
   ConfigEngine.bat setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:
   ```
   '<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql'.
   ```

**Creating users manually**

Use this alternative method for creating users if you have problems running the setup-database task that is documented for setting up a remote SQL Server 2005 database on Windows for a stand-alone production server.

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server

**Previous topic:** Modifying database properties

**Next topic:** Solaris stand-alone server: Assigning custom filegroups on SQL Server 2005
Creating users manually

Use this alternative method for creating users if you have problems running the setup-database task that is documented for setting up a remote SQL Server 2005 database on Windows for a stand-alone production server.

Before you begin:

- You should have completed installing SQL Server 2005 and creating databases.

- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

Create the recommended database users with the SQL Server Management Studio. At least one user is required for each SQL Server 2005 instance.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Connect to your SQL Server 2005 instance.
2. Expand the tree view beneath the SQL Server instance.
3. Expand Security and right-click on Logins.
4. In the opening context menu, select New Login....
5. Enter the database user names.
7. Set a password for the selected user.
8. In the Database Access window, select the database the user must connect to at runtime. The following mappings are recommended: Table 1. List of mappings by database

<table>
<thead>
<tr>
<th>Database</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELEASE</td>
<td>RELEASEUSR</td>
</tr>
<tr>
<td>COMMUNITY</td>
<td>COMMUNITYUSR</td>
</tr>
<tr>
<td>CUSTOMIZATION</td>
<td>CUSTOMIZATIONUSR</td>
</tr>
<tr>
<td>WORKFLOW</td>
<td>WORKFLOWUSR</td>
</tr>
<tr>
<td>WMM</td>
<td>WMMDBUSR</td>
</tr>
<tr>
<td>JCRDB</td>
<td>JCR</td>
</tr>
<tr>
<td>FDBKDB</td>
<td>FEEDBACK</td>
</tr>
<tr>
<td>LMDB</td>
<td>LIKEMINDS</td>
</tr>
</tbody>
</table>

9. Click OK to save the user changes.
10. Add the role SqlJDBCXAUser to all the database users that connect to the database using XA connections. This role is available for the database "master" only. To grant users this role:

   A. Open the Microsoft SQL Server Management Studio and connect to the database instance.
   B. Expand Security > Logins underneath the database instance name.
   C. Click the user name to open the Login Properties window.
   D. Select User Mapping.
E. Select database master in the database list.
F. Select SqlJDBCXAUser.
G. Click OK.

Parent topic: Setting up databases
Solaris stand-alone server: Assigning custom filegroups on SQL Server 2005

The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

Before you begin:
- The custom filegroups must exist prior to the execution of the database transfer.
- To see which table spaces can be customized in each domain, refer to the wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating filegroups refer to the documentation for the database.

**Prerequisites**
- Installing SQL Server
- Modifying database properties
- Setting up databases

If custom filegroups are assigned, each must be assigned explicitly. The default filegroups can be used to contain database objects; however the name of the default file group must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom filegroups:

1. Determine the names of your custom filegroups.
2. Open the mapping file wp_profile_root/PortalServer/config/tablespaces that specifies the table space and index space for each property pairs for each database table:
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.indexspace.indexspace
   
   For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds

3. Assign a filespace to each entry in the mapping file. The filegroup name must be prepended by the keyword ON and a space. For example: community.COMP_INST.tablespace=ON COMM8KSPACE
   
   Repeat this step for each domain that you are transferring.

4. Save and close dbdomain.space_mapping.properties.

5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
   - UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server  
**Previous topic:** Setting up databases  
**Next topic:** Configuring WebSphere Portal to use SQL Server 2005
Configuring WebSphere Portal to use SQL Server 2005

View information on manually transferring data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

Prerequisites
- Installing SQL Server
- Modifying database properties
- Setting up databases

Tips:
- If you are transferring from Oracle, the open Cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

Steps for transferring data to another supported database

1. Open a command prompt and change to the directory wp_profile_root\ConfigEngine.
2. Enter the following commands to validate configuration properties.
   ConfigEngine.bat validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ConfigEngine.bat validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
3. From the same command prompt as the previous steps, change to the directory wp_profile_root\bin.
4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td>stopServer.bat server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td>stopServer.bat WebSphere_Portal -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

5. Transfer the database. **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.
   A. Change to the directory wp_profile_root\ConfigEngine.
Enter the following command:

```bash
ConfigEngine.bat database-transfer -DWasPassword=password
```

**Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:

```bash
ConfigEngine.bat database-transfer -DTransferDomainList=jcr -DWasPassword=password
```

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. Change to the directory `wp_profile_root\bin`.

7. Enter the following command to start the WebSphere Portal server:

```bash
startServer.bat WebSphere_Portal
```

8. Update the SQL Server 2005 statistics for Portal, and JCR databases by opening SQL Server Management Studio, selecting **New Query**, and running the following query:

```sql
use db_name
exec sp_updatestats @resample='resample';
```

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server

**Previous topic:** Solaris stand-alone server: Assigning custom filegroups on SQL Server 2005
Verifying databases

After you configure IBM® WebSphere® Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.

- Prerequisites
  - [Technotes for database connectivity issues](#)

You can verify the database connection using IBM WebSphere Application Server or by opening WebSphere Portal in a browser.

- To verify that the WebSphere Portal application server is running by using WebSphere Application Server, complete these steps:
  1. Open the WebSphere Application Server administrative console by entering the following address in a browser:
     
     http://hostname.example.com:10027/ibm/console
     
     where hostname.example.com is the fully qualified host name of the machine where WebSphere Portal is running and 10027 is the default transport port that is created by WebSphere Application Server.
  2. Log into the administrative console.
  3. Depending on your version of WebSphere Application Server, click the appropriate option:
     - For WebSphere Application Server Version 6.1: Click Resources & JDBC Providers.
     - For WebSphere Application Server Version 7.0: Click Resources > JDBC > JDBC Providers
  4. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
  5. Select the name of the data source that is defined in wkplc_comp.properties. The default data source is wpdbDS.
  6. Select the name of the JDBC provider that is specified in wkplc_dbtype.properties. The default JDBC provider is wpdbJDBC_dbtype, where dbtype is replaced by the value that matches your environment.
  7. Click Test Connection to verify the database connection. If configuration parameters have been changed, you might need to restart WebSphere Application Server for the test to complete.

- To verify that the WebSphere Portal application server is running by opening WebSphere Portal in a browser, enter the following URL in a supported browser:

http://hostname.example.com:10040/wps/portal

where hostname.example.com is the fully qualified host name of the machine where WebSphere Portal is running and 10040 is the default transport port that is created by WebSphere Application Server.

Parent topic: Configuring databases
Preparing a remote Web server when portal is installed on Solaris

Install and configure the Web server plug-in, provided by IBM® WebSphere® Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

- **Prerequisites**
  - Configuring databases

Perform the following steps to install and configure your Web server:

1. Install and configure the Web server; refer to the Web server documentation for information.
2. If using IBM Lotus® Domino®, edit the `NOTES.INI` file on the Web server. Set the `HTTPEnableConnectorHeaders` parameter to 1.
3. If using IBM HTTP Server or Apache Server, edit the `httpd.conf` file on the Web server. Set the `AllowEncodedSlashes` directive to `on`; the directive should be added at the root level as a global directive.

4. Stop the Web server.
5. Install and configure the Web server plug-in on the system where the Web server is located using the Plug-ins installation wizard provided with WebSphere Application Server. Refer to Selecting a Web server topology diagram and roadmap for information. **Note:** The WebSphere Application Server Web server instructions are for the default profile. If not using the default profile, refer to Configuring a Web server for a non-default profile.

   **If using WebDAV:** After successfully installing the Web server plug-in, locate and open your `plugin-cfg.xml` file and set `AcceptAllContent` to `true`.

   **Important:** Depending on how you use the Web server, you may need to adjust the `ServerIOTimeout` value, which defines how long the plug-in should wait for a response from the application. The recommended minimum value is 60 but you may need to adjust this value higher if you are retrieving data from a database. To update this value, locate and open your `plugin-cfg.xml` file and set `ServerIOTimeout` to a value that is appropriate for your business needs. For additional information, see Common questions about the Web server plug-in.

6. If you are using Sun Web Server Version 7 update 8, read Technote 1448262 and perform the steps to resolve the HTTP 408/409 error.
7. Start the Web server.

**Parent topic:** Setting up a stand-alone server on Solaris  
**Previous topic:** Configuring databases  
**Next topic:** Configuring WebSphere Portal to use a user registry on Solaris
Configuring WebSphere Portal to use a user registry on Solaris

Configure user registry security on IBM® WebSphere® Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

Prior to configuring security, you should use the IBM WebSphere Application Server backupConfig task to create and store a backup of the IBM WebSphere Portal configuration; see backupConfig command for information.

- Prerequisites
  - Configuring databases

Perform the following tasks to configure WebSphere Portal to use a user registry:

1. Preparing user registries on Solaris
   Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

2. Choosing your user registry model on Solaris
   Choose between securing IBM WebSphere Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

3. Adapting the attribute configuration
   After installing IBM WebSphere Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

4. Stand-alone server: Configuring WebSphere Portal to use dynamic groups on Solaris
   By default, WebSphere Portal is enabled for static groups. However, the Virtual Member Manager (VMM) allows users to be members of either static or dynamic groups. Static groups are those where a persistent binding exists between a group and its members. Dynamic groups are those where a search query is defined to retrieve the members of a group. If you have your LDAP server configured to use dynamic groups, complete the steps in this task for WebSphere Portal to use dynamic group queries when you setup your LDAP server.

5. Stand-alone server: Enabling referrals for your LDAP user registry on Solaris
   Referrals redirect object requests from one LDAP server to another when objects do not exist or cannot be located in a particular directory tree. You should enable referrals if your environment has more than one user registry existing on multiple servers or domains.

Parent topic: Setting up a stand-alone server on Solaris
Previous topic: Preparing a remote Web server when portal is installed on Solaris
Next topic: Tune your servers

Related tasks
Managing your user registry on Solaris
Preparing user registries on Solaris
Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

Choose the appropriate LDAP server to install and setup:

- **Preparing a Domino Directory server**
  If you plan to use a Domino Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a SecureWay Security Server**
  If you plan to use a SecureWay Security Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Novell eDirectory**
  If you plan to use a Novell eDirectory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Sun Java System Directory Server**
  If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Tivoli Directory Server**
  If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

Parent topic: Configuring WebSphere Portal to use a user registry on Solaris
Next topic: Choosing your user registry model on Solaris
Preparing a Domino Directory server

If you plan to use a Domino Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Domino Directory:

1. Perform the following steps to install the Domino Directory:
   A. Go to Lotus Domino documentation.
   B. Select the appropriate version tab for your product.
   C. Click the Download/View online link for the Lotus Domino for multiple platforms Information Center.
   D. Click Domino Administrator Help > Installation > Installing and setting up Domino servers > Server installation > Installing Domino on Unix and perform this task.
   E. Click Domino Administrator Help > Installation > Installing and setting up Domino servers > The Domino server setup program and perform this task.

2. Perform the following steps as a guide to create the WebSphere Portal administrative user:
   A. Navigate to the People view of the Domino Directory and then click Add Person.
   B. Enter the following values in the New Person form to create the wpsbind user:
      - Last Name
        - wpsbind
      - User Name
        - wpsbind/DominoDomain, where DominoDomain is your Lotus Domino Internet domain
        - wpsbind Note: Make sure you enter two values in the User Name field, where the first value includes the Lotus Domino domain.
      - Short name/UserID
        - wpsbind
      - Internet password
        - wpsbind
   C. Click Save and Close to save the new person record for wpsbind and return to the People view.
   D. Click Add Person and enter the following values in the New Person form to create the wpsadmin user:
      - Last Name
        - wpsadmin, where wpsadmin in the user ID for the WebSphere Portal Administrator
      - User Name
        - wpsadmin/DominoDomain, where DominoDomain is your Lotus Domino Internet domain
        - wpsadmin Note: Make sure you enter two values in the User Name field, where the first value includes the Lotus Domino domain.
      - Short name/UserID
        - wpsadmin
      - Internet password
        - wpsadmin
   E. Click Save and Close to save the new person record for wpsadmin and return to the People view.
   F. Navigate to the Groups view and click Add Group.
   G. Enter the following values in the New Group form on the Basic tab:
      - Group name
Note: If you plan to configure WebSphere Portal for multiple user registries and your Lotus Domino LDAP will share a realm with another user registry, you must use the hierarchical naming convention for the group names, for example: wpsadmins/DominoDomain, to avoid unexpected results during WebSphere Portal runtime.

**- Group type**
- Multi-purpose

**- Members**
- wpsbind/DominoDomain
  - wpsadmin/DominoDomain

Note: You can add additional administrator users if required.

- Click **Save and Close** to save the wpsadmins group with the wpsbind and wpsadmin users as members.

3. Perform the following steps to update the access control list for the Domino Directory:
   
   A. Open the names.nsf file in the Lotus Domino Administrator or Lotus Notes client.
   B. Click **File > Application > Access Control** from the main menu to open the access control list for the file.
   C. In the **Access Control List > Basics** panel, ensure that the wpsadmins group has either Author or Editor access.
   D. Add the following Role Types to the wpsadmins group:
      - GroupCreator
      - GroupModifier
      - UserCreator
      - UserModifier
   E. Click **OK**.

Parent topic: Preparing user registries on Solaris
Preparing a SecureWay Security Server

If you plan to use a SecureWay Security Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare SecureWay Security Server:

1. Install SecureWay Security Server; refer to the IBM SecureWay™ Security Server for z/OS® and OS/390® for information.

2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every dc=yourco,dc=com with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupId.
   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on Solaris
Preparing a Novell eDirectory

If you plan to use a Novell eDirectory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Novell eDirectory:

1. Install Novell eDirectory; refer to eDirectory for information.
2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every dc=yourco,dc=com with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.
   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on Solaris
Preparing a Sun Java System Directory Server

If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Sun Java System Directory Server:

1. Install Sun Java System Directory Server; refer to Oracle software for information.
2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every dc=yourco,dc=com with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.
   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on Solaris
Preparing a Tivoli Directory Server

If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Tivoli Directory Server:

   
   **Note:** Due to a restriction in Tivoli Directory Server, users or groups must not contain a Turkish uppercase dotted I or lowercase dotted i in the DN as this will prevent correct retrieval of that user or group.

2. Perform the following steps to create the WebSphere Portal administrative user:
   
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the **Server Administration** folder, located in the left-hand navigation of the directory server console.
      2. Click the **Manage Server Properties** folder under the Server Administration folder and then select **Suffixes** on the main page.
      3. Type the **Base DN** name for the suffix; for example: `dc=yourcompany,dc=com`.
      4. Click **Add**.
      5. Click **OK** to save your changes.
      6. Stop and restart the LDAP server.
   
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the `PortalUsers.ldif` file as a working example and adapted appropriately to work with your LDAP server.
      - Use the `ContentUsers.ldif` file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.

   C. Replace every `dc=yourco,dc=com` with your suffix.

   D. Replace any prefixes and suffixes that are unique to your LDAP server.

   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.

   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the **objectclasses** to **accessGroup**. If using Tivoli Access Manager Version 6, set the **objectclasses** to **groupOfNames**.

   G. Save your changes.

   H. Follow the instructions provided with your directory server to import the LDIF file.

   I. Stop and restart the LDAP server.

**Parent topic:** Preparing user registries on Solaris
Choosing your user registry model on Solaris

Choose between securing IBM® WebSphere® Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

Choose one of the following user registry models:

- **Configuring a stand-alone LDAP user registry on Solaris**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Configuring the default federated repository on Solaris**
  Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

Parent topic: Configuring WebSphere Portal to use a user registry on Solaris

Previous topic: Preparing user registries on Solaris

Next topic: Adapting the attribute configuration
Configuring a stand-alone LDAP user registry on Solaris

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Configuring a stand-alone LDAP user registry on Solaris**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry to store all user account information for authorization.

- **Configuring a stand-alone LDAP user registry over SSL on Solaris**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

Parent topic: Choosing your user registry model on Solaris
Configuring a stand-alone LDAP user registry on Solaris

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry to store all user account information for authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps. If you need to rerun the \textit{wp-modify-ldap-security} task to change the LDAP repositories or because the task failed, you must choose a new name for the realm using the \texttt{standalone.ldap.realm} parameter or you can set \texttt{ignoreDuplicateIDs=true} in the \texttt{wkplc.properties} file, before rerunning the task.

Perform the following steps to configure a standalone LDAP user registry:

\textbf{Note:} Use the \texttt{wp_security_xxx.properties} helper file, located in the \texttt{wp_profile_root}/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the \texttt{wkplc.properties} file, you will use your \texttt{wp_security_xxx.properties} helper file.

1. Use a text editor to open the \texttt{wkplc.properties} file, located in the \texttt{wp_profile_root}/ConfigEngine/properties directory.

2. Required: Enter a value for the following required parameters in the \texttt{wkplc.properties} file under the VMM Stand-alone LDAP configuration heading:

   \textbf{Note:} See the properties file for specific information about the required parameters and for advanced parameters.

   - \texttt{standalone.ldap.id}
   - \texttt{standalone.ldap.host}
   - \texttt{standalone.ldap.port}
   - \texttt{standalone.ldap.bindDN}
   - \texttt{standalone.ldap.bindPassword}
   - \texttt{standalone.ldap.ldapServerType}
   - \texttt{standalone.ldap.userIdMap}
   - \texttt{standalone.ldap.groupIdMap}
   - \texttt{standalone.ldap.groupMemberIdMap}
   - \texttt{standalone.ldap.userFilter}
   - \texttt{standalone.ldap.groupFilter}
   - \texttt{standalone.ldap.serverId}
   - \texttt{standalone.ldap.serverPassword}
   - \texttt{standalone.ldap.realm}
   - \texttt{standalone.ldap.primaryAdminId}
   - \texttt{standalone.ldap.primaryAdminPassword}
   - \texttt{standalone.ldap.primaryPortalAdminId}
   - \texttt{standalone.ldap.primaryPortalAdminPassword}
   - \texttt{standalone.ldap.primaryPortalAdminGroup}
   - \texttt{standalone.ldap.baseDN}
3. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading:

- `standalone.idap.et.group.objectClasses`
- `standalone.idap.et.group.objectClassesForCreate`
- `standalone.idap.et.group.searchBases`
- `standalone.idap.et.personaccount.objectClasses`
- `standalone.idap.et.personaccount.objectClassesForCreate`
- `standalone.idap.et.personaccount.searchBases`

**Note:** See the properties file for specific information about the required parameters and for advanced parameters.

4. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attributes heading:

- `standalone.ldap.gm.groupMemberName`
- `standalone.ldap.gm.objectClass`
- `standalone.ldap.gm.scope`
- `standalone.ldap.gm.dummyMember`

**Note:** See the properties file for specific information about the required parameters and for advanced parameters.

5. Required: Enter a value for the following required relative distinguished name (RDN®) parameters in the `wkplc.properties` file under the Default parent, RDN attribute heading:

- `standalone.idap.personAccountParent`
- `standalone.idap.groupParent`
- `standalone.idap.personAccountRdnProperties`
- `standalone.idap.groupRdnProperties`

**Note:** See the `wkplc.properties` file for specific information about the required parameters and for advanced parameters.

6. Save your changes to the `wkplc.properties` file.

7. Optional: Enter the following Web content authors parameters in the `wkplc_comp.properties` file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product:

- `WcmContentAuthorsGroupId`
- `WcmContentAuthorsGroupCN`

**Note:** See the `wkplc_comp.properties` file for specific information about the required parameters and for advanced parameters.

8. Run the `./ConfigEngine.sh validate-standalone-ldap -DWasPassword=password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, `WasPassword` is the value entered during installation and not a value found in your LDAP user registry.

**Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now? Press y then Enter.

9. Run the `./ConfigEngine.sh wp-modify-ldap-security -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to set the stand-alone LDAP user registry.

10. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

11. Run the `./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

12. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the
LDAP directory.

**Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

A. Edit the `wp_profile_root` 

`/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties` file.

B. Add the following lines to the file:

```
uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
```

Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `./ConfigEngine.sh action=express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.idap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

13. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

**Parent topic:** Configuring a stand-alone LDAP user registry on Solaris

**Related tasks**

- [Adapting the attribute configuration](#)
- [Using the member fixer tool with IBM Lotus Web Content Management](#)
- [Starting and stopping servers, deployment managers, and node agents](#)
Configuring a stand-alone LDAP user registry over SSL on Solaris

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to configure a standalone LDAP user registry over SSL:

**Note:** Use the `wp_security_xxx.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_security_xxx.properties` helper file.

1. To specify the LDAP server’s SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:
   - **Server trust store**
     A. Add the certificate to the trust store:
        1. Log in to the WebSphere Application Server Administrative Console.
        2. Navigate to **Security > SSL certificate and key management > SSL configurations**.
        3. Click the appropriate SSL configuration from the list. For example,
           - Stand-alone environments: `NodeDefaultSSLSettings`
           - Clustered environments: `CellDefaultSSLSettings`
           **Clustered environments:** Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.
        4. Click **Key stores and certificates**.
        5. Click the appropriate trust store from the list; for example, `NodeDefaultTrustStore`.
        6. Click **Signer certificates**, click **Add**, and then enter the following information:
           - Type the **Alias** the key store uses for the signer certificate.
           - Type the **File name** where the signer certificate is located.
       7. Click **OK** and then click **Save** to save the changes to the master configuration.

   B. Retrieve the certificate from the port:
      1. Log in to the WebSphere Application Server Administrative Console.
      2. Navigate to **Security > SSL certificate and key management > SSL configurations**.
      3. Click the appropriate SSL configuration from the list. For example,
         - Stand-alone environments: `NodeDefaultSSLSettings`
- Clustered environments: **CellDefaultSSLSettings**

Clustered environments: Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.

4. Click **Key stores and certificates**.

5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.

6. Click **Signer certificates**, click **Retrieve from port**, and then enter the following information:
   - Type the **Host** name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the **Port** used when attempting to retrieve the signer certificate.
   - Type the **Alias** the key store uses for the signer certificate.

7. Click **Retrieve signer information** to retrieve the certificate from the port.

8. Click **OK** and then click **Save** to save the changes to the master configuration.

- **Client trust storeNote:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

A. See Secure installation for client signer retrieval.

B. Run the retrieveSigners task from the **wp_profile_root/bin** directory; see retrieveSigners command for information. In a deployed environment, you will need to run the retrieveSigners task, for any federated node, against the Deployment Manager. **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

C. Update the trust store properties file:

1. Use a text editor to open the **ssl.client.props** file, located in the **wp_profile_root/properties** directory.

2. Change the **com.ibm.ssl.trustStore** parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter
   
   ```
   com.ibm.ssl.trustStore=${CONFIG_ROOT}\cells\wpsbvt\nodes\wpsbvt\trust.p12
   ```
   
   to use the default trust store.

3. Save your changes.

2. Use a text editor to open the **wkplc.properties** file, located in the **wp_profile_root/ConfigEngine/properties** directory.

3. Required: Enter a value for the following required parameters in the **wkplc.properties** file under the VMM Standalone LDAP configuration heading. **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - standalone.idap.id
   - standalone.idap.host
   - standalone.idap.port
   - standalone.idap.blnDDN
   - standalone.idap.bindPassword
   - standalone.idap.idapServerType
   - standalone.idap.userIdMap
   - standalone.idap.groupldMap
   - standalone.idap.groupMemberIdMap
   - standalone.idap.userFilter
   - standalone.idap.groupFilter
   - standalone.idap.serverId
   - standalone.idap.serverPassword
   - standalone.idap.realm
   - standalone.idap.primaryAdminId
   - standalone.idap.primaryAdminPassword
   - standalone.idap.primaryPortalAdminId
4. Required: Enter a value for the following required entity types parameters in the wkplc.properties file under the LDAP entity types heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.et.group.objectClasses
   - standalone.ldap.et.group.objectClassesForCreate
   - standalone.ldap.et.group.searchBases
   - standalone.ldap.et.personaccount.objectClasses
   - standalone.ldap.et.personaccount.objectClassesForCreate
   - standalone.ldap.et.personaccount.searchBases

5. Required: Enter a value for the following required group member parameters in the wkplc.properties file under the Group member attributes heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.gm.groupMemberName
   - standalone.ldap.gm.objectClass
   - standalone.ldap.gm.scope
   - standalone.ldap.gm.dummyMember

6. Required: Enter a value for the following required relative distinguished name (RDN®) parameters in the wkplc.properties file under the Default parent, RDN attribute heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.personAccountParent
   - standalone.ldap.groupParent
   - standalone.ldap.personAccountRdnProperties
   - standalone.ldap.groupRdnProperties

7. Enter a value for the following parameters to enable Secure Socket Layers (SSL): Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.sslEnabled
   - standalone.ldap.sslConfiguration

Optional parameters:
   - standalone.ldap.certificateMapMode
   - standalone.ldap.certificateFilter

8. Save your changes to the wkplc.properties file.

9. Run the ./ConfigEngine.sh validate-standalone-ldap -DWasPassword=password task to validate your LDAP server settings. Attention: If you have not deleted the default file repository, WasPassword is the value entered during installation and not a value found in your LDAP user registry.

   Note: During the validation task, you may receive the following prompt: Add signer to the trust store now?.

   Press y then Enter.

10. Run the ./ConfigEngine.sh wp-modify-ldap-security -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to set the stand-alone LDAP user registry.

11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

12. Run the ./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to check that all defined attributes are available in the configured
LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

**Parent topic:** Configuring a stand-alone LDAP user registry on Solaris

**Related tasks**
- Adapting the attribute configuration
- Starting and stopping servers, deployment managers, and node agents
Configuring the default federated repository on Solaris

Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

Perform the following tasks to add user registries to the default federated repository:

- **Configuring a federated LDAP user registry on Solaris**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Adding a database user registry on Solaris**
  Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

- **Adding realm support on Solaris**
  A realm is a group of users from one or more user registries that form a coherent group within IBM WebSphere Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

**Parent topic:** Choosing your user registry model on Solaris
Configuring a federated LDAP user registry on Solaris

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Adding an LDAP user registry on Solaris**
  Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time. If IBM Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=groupName` and the hierarchical format is `cn=groupName,o=root`. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as `wpsadmin`, this ID cannot exist in the LDAP you are adding.

- **Adding an LDAP user registry over SSL on Solaris**
  Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

Parent topic: Configuring the default federated repository on Solaris
Adding an LDAP user registry on Solaris

Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

If IBM® Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is cn=groupName and the hierarchical format is cn=groupName, o=root. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as wpsadmin, this ID cannot exist in the LDAP you are adding.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
2. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM Federated LDAP Properties heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - federated.ldap.id
   - federated.ldap.host
   - federated.ldap.port
   - federated.ldap.bindDN
   - federated.ldap.bindPassword
   - federated.ldap.ldapServerType
   - federated.ldap.baseDN
3. Required: Enter a value for the following required entity types parameters in the wkplc.properties file under the LDAP entity types heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - federated.ldap.et.group.objectClasses
   - federated.ldap.et.group.objectClassesForCreate
   - federated.ldap.et.group.searchBases
   - federated.ldap.et.personaccount.objectClasses
   - federated.ldap.et.personaccount.objectClassesForCreate
   - federated.ldap.et.personaccount.searchBases
4. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attribute heading. **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `federated.ldap.gm.groupMemberName`
   - `federated.ldap.gm.objectClass`
   - `federated.ldap.gm.scope`
   - `federated.ldap.gm.dummyMember`

5. Save your changes to the `wkplc.properties` file.

6. Optional: Enter the following Web content authors parameters in the `wkplc_comp.properties` file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. **Note:** See the `wkplc_comp.properties` file for specific information about the required parameters and for advanced parameters.
   - `WcmContentAuthorsGroupId`
   - `WcmContentAuthorsGroupCN`

7. Run the `.ConfigEngine.sh validate-federated-ldap -DWasPassword=password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, `WasPassword` is the value entered during installation and not a value found in your LDAP user registry. **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now? Press `y` then Enter.

8. Run the `.ConfigEngine.sh wp-create-ldap -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to add an LDAP user registry to the default federated repository. **Note:** Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

10. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:
    A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
    B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
       - `id`
       - `baseDN`
       - `nameInRepository`
    C. Save your changes to the `wkplc.properties` file.
    D. Run the `.ConfigEngine.sh wp-create-base-entry -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to create a base entry in a repository.
    E. Stop and restart all necessary servers to propagate your changes.

11. Optional: Run the `.ConfigEngine.sh wp-query-repository -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to list the names and types of configured repositories.

12. Run the `.ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between
WebSphere Portal and the LDAP server.

13. Perform the following steps to update the user registry where new users and groups are stored:

   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - `personAccountParent`
      - `groupParent`
      - `personAccountRdnProperties`
      - `groupRdnProperties`

      The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:

      - `personAccountParent=dc=yourco,dc=com`
      - `groupParent=dc=yourco,dc=com`

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to delete the old attributes before adding the new attributes.

   E. Stop and restart all necessary servers to propagate your changes.

14. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for `realmName` or leave blank to update the default realm.

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

   E. Stop and restart all necessary servers to propagate your changes.

15. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

   **Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

   A. Edit the `wp_profile_root/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties` file.

   B. Add the following lines to the file:

   ```properties
   uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
   cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
   ```

   Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.
Save your changes and close the file.

Run the `./ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=passw0rd -DWasPassword=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. *Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management*

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.idap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

16. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

17. Optional: **This step is required in a production environment.** Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

   **Important:**
   - Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
   - Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

   **Note:** If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

   A. Run the following task from the `wp_profile_root/ConfigEngine` directory: `./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DNwAdminId=newadminId -DNwAdminPw=newpassword` **Important:** You must provide the full distinguished name (DN) for the `newAdminId` parameter.

   B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

   C. Run the following task from the `wp_profile_root/ConfigEngine` directory: `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DNwAdminId=newadminId -DNwAdminPw=newpassword -DNwAdminGroupId=newadmingroupid` **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

   D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

18. Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it.

   The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by
duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: *Deleting the repository.*

**Parent topic:** Configuring a federated LDAP user registry on Solaris

**Related tasks**
Adapting the attribute configuration
Deleting the repository on Solaris
Starting and stopping servers, deployment managers, and node agents

**Related information**
User IDs and passwords
Adding an LDAP user registry over SSL on Solaris

Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry over SSL to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

**Note:** Use the *wp_add_federated_xxx.properties* helper file, located in the *wp_profile_root/ConfigEngine/config/helpers* directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the *wkplc.properties* file, you will use your *wp_add_federated_xxx.properties* helper file.

1. To specify the LDAP server's SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:
   - **Server trust store**
     A. Add the certificate to the trust store:
        1. Log in to the WebSphere Application Server Administrative Console.
        2. Navigate to **Security > SSL certificate and key management > SSL configurations**.
        3. Click the appropriate SSL configuration from the list. For example,
           - Stand-alone environments: *NodeDefaultSSLSettings*
           - Clustered environments: *CellDefaultSSLSettings*
    **Clustered environments:** Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.
        4. Click **Key stores and certificates**.
        5. Click the appropriate trust store from the list; for example, *NodeDefaultTrustStore*.
        6. Click **Signer certificates**, click **Add**, and then enter the following information:
           - Type the **Alias** the key store uses for the signer certificate.
           - Type the **File name** where the signer certificate is located.
        7. Click **OK** and then click **Save** to save the changes to the master configuration.
   B. Retrieve the certificate from the port:
      1. Log in to the WebSphere Application Server Administrative Console.
      2. Navigate to **Security > SSL certificate and key management > SSL configurations**.
      3. Click the appropriate SSL configuration from the list. For example, 
         - Stand-alone environments: *NodeDefaultSSLSettings*
Clustered environments: **CellDefaultSSLSettings**

**Clustered environments:** Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.

4. Click **Key stores and certificates**.

5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.

6. Click **Signer certificates**, click **Retrieve from port**, and then enter the following information:
   - Type the **Host** name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the **SSL Port** used when attempting to retrieve the signer certificate.
   - Type the **Alias** the key store uses for the signer certificate.

7. Click **Retrieve signer information** to retrieve the certificate from the port.

8. Click **OK** and then click **Save** to save the changes to the master configuration.

**Client trust store**Note: This task might report an error, but it does successfully update the trust store. You can ignore the error message.

A. See Secure installation for client signer retrieval.

B. Run the `retrieveSigners` task from the `wp_profile_root/bin` directory; see `retrieveSigners` command for information. In a deployed environment, you will need to run the `retrieveSigners` task, for any federated node, against the Deployment Manager.**Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

C. Update the trust store properties file:

   1. Use a text editor to open the `ssl.client.props` file, located in the `wp_profile_root/properties` directory.

   2. Change the **com.ibm.ssl.trustStore** parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter
      `com.ibm.ssl.trustStore=\{CONFIG_ROOT\}/cells\wpsbvt\nodes\wpsbvt\trust.p12` to use the default trust store.

   3. Save your changes.

2. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

3. Required: Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated LDAP Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - `federated.ldap.id`
   - `federated.ldap.host`
   - `federated.ldap.port`
   - `federated.ldap.bindDN`
   - `federated.ldap.bindPassword`
   - `federated.ldap.ldapServerType`
   - `federated.ldap.baseDN`

4. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading:**Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - `federated.ldap.et.group.objectClasses`
   - `federated.ldap.et.group.objectClassesForCreate`
   - `federated.ldap.et.group.searchBases`
   - `federated.ldap.et.personaccount.objectClasses`
   - `federated.ldap.et.personaccount.objectClassesForCreate`
   - `federated.ldap.et.personaccount.searchBases`
5. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attribute heading:

- `federated.idap.gm.groupMemberName`
- `federated.idap.gm.objectClass`
- `federated.idap.gm.scope`
- `federated.idap.gm.dummyMember`

6. Enter a value for the following parameters to enable Secure Socket Layers (SSL):

- `federated.idap.sslEnabled`
- `federated.idap.sslConfiguration`

Optional parameters:

- `federated.idap.certificateMapMode`
- `federated.idap.certificateFilter`

7. Save your changes to the `wkplc.properties` file.

8. Optional: Enter the following Web content authors parameters in the `wkplc_comp.properties` file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product:

- `WcmContentAuthorsGroupId`
- `WcmContentAuthorsGroupCN`

9. Run the `./ConfigEngine.sh validate-federated-ldap -DWasPassword= password` task to validate your LDAP server settings.

   **Attention:** If you have not deleted the default file repository, `WasPassword` is the value entered during installation and not a value found in your LDAP user registry.

   **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now? Press **y** then **Enter**.

10. Run the `./ConfigEngine.sh wp-create-ldap -DWasPassword= password` task, from the `wp_profile_root/ConfigEngine` directory, to add an LDAP user registry to the default federated repository.

   **Note:** Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

12. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms:

   **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - `id`
   - `baseDN`
Save your changes to the `wkplc.properties` file.

Run the `./ConfigEngine.sh wp-create-base-entry -DWasPassword=password` task, from the `wp_profile_root` directory, to create a base entry in a repository.

Stop and restart all necessary servers to propagate your changes.

Optional: Run the `./ConfigEngine.sh wp-query-repository -DWasPassword=password` task, from the `wp_profile_root` directory, to list the names and types of configured repositories.

Run the `./ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root` directory, to check that all defined attributes are available in the configured LDAP user registry. Important: When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

Perform the following steps to update the user registry where new users and groups are stored:

Note: If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

Attention: During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

Note: See the properties file for specific information about the required parameters and for advanced parameters.

- `personAccountParent`
- `groupParent`
- `personAccountRdnProperties`
- `groupRdnProperties`

The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:

- `personAccountParent=dc=yourco,dc=com`
- `groupParent=dc=yourco,dc=com`

C. Save your changes to the `wkplc.properties` file.

D. Run the `./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root` directory, to delete the old attributes before adding the new attributes.

E. Stop and restart all necessary servers to propagate your changes.

Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. Enter a value for `realmName` or leave blank to update the default realm.

C. Save your changes to the `wkplc.properties` file.

D. Run the `./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. Note: After running this task to enable the full distinguished name login, you can run the `./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

E. Stop and restart all necessary servers to propagate your changes.

Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the
Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

**Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

A. Edit the `wp_profile_root`/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties file.

B. Add the following lines to the file:

```
uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
```

Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `./ConfigEngine.sh` action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password task, located in the `wp_profile_root/ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

18. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

19. Optional: **This step is required in a production environment.** Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

**Important:**
- Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
- Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

**Note:** If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

A. Run the following task from the `wp_profile_root/ConfigEngine` directory: `./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword`. **Important:** You must provide the full distinguished name (DN) for the `newAdminid` parameter.

B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.
C. Run the following task from the `wp_profile_root/ConfigEngine` directory:
```
./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid
```
**Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it. The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: *Deleting the repository.*

Parent topic: Configuring a federated LDAP user registry on Solaris

Related tasks
- Adapting the attribute configuration
- Deleting the repository on Solaris
- Starting and stopping servers, deployment managers, and node agents

Related information
- User IDs and passwords
Adding a database user registry on Solaris

Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add a database user registry to the default federated repository; you must repeat these steps for each additional database user registry that you plan to add:

Note: Use the wp_add_DB.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wpcl.properties file, you will use your wp_add_DB.properties helper file.

1. Set up a new database, including creating a new user with appropriate database privileges for accessing the database:

   **Instructions for setting up databases:** Refer to the appropriate documentation for the type of database you want to set up.

   **Consulting your database administrator:** The task of setting up a new database is typically performed by a database administrator. However, the following steps are provided for your reference in the event you are creating a stand-alone database for testing or demonstration purposes. Consult your database administrator before proceeding with the following steps if you plan to create a database for a production environment.

   **Table 1. Steps for creating a new database to use as a database user registry.**

<table>
<thead>
<tr>
<th>Database</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td>Perform the following steps to create a DB2 database: Install DB2. Enter the following database tuning commands: ( \text{db2} ) &quot;CREATE DB ( \text{dbname} ) using codeset UTF-8 territory us PAGESIZE 8192&quot; ( \text{db2} ) &quot;UPDATE DB CFG FOR ( \text{dbname} ) USING applheapsz 4096&quot; ( \text{db2} ) &quot;UPDATE DB CFG FOR ( \text{dbname} ) USING app_ctl_heap_sz 1024&quot; ( \text{db2} ) &quot;UPDATE DB CFG FOR ( \text{dbname} ) USING stmtimeap 32768&quot; ( \text{db2} ) &quot;UPDATE DB CFG FOR ( \text{dbname} ) USING dbheap 2400&quot; ( \text{db2} ) &quot;UPDATE DB CFG FOR ( \text{dbname} ) USING locklist 1000&quot; ( \text{db2} ) &quot;UPDATE DB CFG FOR ( \text{dbname} ) USING logfilsiz 4000&quot; ( \text{db2} ) &quot;UPDATE DB CFG FOR ( \text{dbname} ) USING logprimary 12&quot; ( \text{db2} ) &quot;UPDATE DB CFG FOR ( \text{dbname} ) USING logsecond 20&quot; ( \text{db2} ) &quot;UPDATE DB CFG FOR ( \text{dbname} ) USING logbackup 32&quot; ( \text{db2} ) &quot;UPDATE DB CFG FOR ( \text{dbname} ) USING avg_appls 5&quot; ( \text{db2} ) &quot;UPDATE DB CFG FOR ( \text{dbname} ) USING locktimeout 30&quot; ( \text{db2} ) &quot;UPDATE DB CFG FOR ( \text{dbname} ) using AUTO_MAINT off&quot;</td>
</tr>
</tbody>
</table>
2. Perform the following steps to define the DbDriver and DbLibrary parameter values:

   A. Navigate to the following directory: `wp_profile_root`/ConfigEngine/properties
   B. Locate and open `wkplc_dbtype.properties` with any text editor.
   C. Enter a value for the following parameters under the appropriate database type properties heading:
      - `db_type(DbDriver`
      - `db_type(DbLibrary`
   D. Save your changes.

3. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root`/ConfigEngine/properties directory.

4. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated Database Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - `federated.db.DataSourceName`
   - `federated.dbDbType`
   - `federated.db.DbUrl`
   - `federated.db.Id`
   - `federated.db.baseDN`
   - `federated.db.DbUser`
   - `federated.db.DbPassword`
   - `federated.db.DbName`

5. Change the value for the `com.ibm.SOAP.requestTimeout` parameter to 1000.

   A. Navigate to the following directory: `wp_profile_root`/properties
   B. Locate and open `soap.client.props` with any text editor.
   C. Locate the `com.ibm.SOAP.requestTimeout` parameter and change the value to 1000.
   D. Save and close `soap.client.props`.

6. Perform the following steps in a clustered environment:

   A. Run the `./ConfigEngine.sh wp-prep-vmm-db-secured-environment -DWasPassword=password -DDbDomain=federated.db -Db_type.DbMgrDbLibrary=local path of the database jars on the Deployment`
Manager -DDmgrNodeName=dmgr_node_name task from the wp_profile_root/ConfigEngine directory to create the local Deployment Manager WebSphere variable used to access the database jars. **Note:** The db_type in db_type .DmgrDbLibrary should be set to the type of database you are using, for example db2. The local full path of the database jars on the Deployment Manager should be one of the following options:

- **DB2 Type 2 driver:** db2java.zip
- **DB2 Type 4 driver:** db2jcc.jar;db2jcc_license_cu.jar
- **DB2 for z/OS Type 2 driver:** db2java.zip
- **DB2 for z/OS Type 4 driver:** db2jcc.jar;db2jcc_license_cisuz.jar;db2jcc_javax.jar
- **Oracle:** ojdbc14.jar
- **SQL Server JDBC driver provided by Microsoft:** sqljdbc.jar
- **SQL Server JDBC driver provided by DataDirect:** sqlserver.jar;base.jar;util.jar

B. Run the following task. Include each node name as a comma separated list in the command:

**Running the task:** You do not have to run this task more than once. You can run this task from any node in the cluster.

1. Set the property value for federated.db.DbType in the wkplc.properties file if using a database user registry.
2. Run the ./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=password -DDDbDomain=federated.db -DVmmNodeName=node_name -Ddb_type.NodeDbLibrary=local full path of the database jars task from the wp_profile_root/ConfigEngine directory on each node to create the variable used to access the VMM database jars. **Note:** VmmNodeName is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The db_type in db_type.NodeDbLibrary should be set to the type of database you are using, for example db2.

C. Stop and restart all necessary servers to propagate your changes.

7. Run the ./ConfigEngine.sh wp-create-db -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add a database user registry to the default federated repository. **Note:** Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: **Starting and stopping servers, deployment managers, and node agents.**

9. Perform the following steps to update the user registry where new users and groups are stored:

**Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

**Attention:** During installation, the default file repository creates a default value in the personAccountRdnProperties and groupRdnProperties parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

B. Enter a value for the following required parameters in the wkplc.properties file under the VMM supported entity types configuration heading:

**Note:** See the properties file for specific information about the required parameters and for advanced parameters.
- `personAccountParent`
- `groupParent`
- `personAccountRdnProperties`
- `groupRdnProperties`

The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:
- `personAccountParent=dc=yourco,dc=com`
- `groupParent=dc=yourco,dc=com`

C. Save your changes to the `wkplc.properties` file.

D. Run the `./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root` directory, to delete the old attributes before adding the new attributes.

E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Run the `./ConfigEngine.sh wp-query-repository -DWasPassword=password` task, from the `wp_profile_root` directory, to list the names and types of configured repositories.

Parent topic: Configuring the default federated repository on Solaris

Parent topic: Updating your user registry on Solaris
Adding realm support on Solaris

A realm is a group of users from one or more user registries that form a coherent group within IBM® WebSphere® Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

Before configuring realm support, you must add all LDAP user registries and/or database user registries, that you will use to create a single realm or multiple realms, to the federated repository. If you are going to create multiple realms, you must create all required base entries within your LDAP user registries and/or database user registries. All base entry names must be unique within the federated repository. In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add realm support to your user registry model:

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
2. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - realmName
   - securityUse
   - delimiter
   - addBaseEntry
3. Save your changes to the wkplc.properties file.
4. Run the ./ConfigEngine.sh wp-create-realm -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add a new realm to the Virtual Member Manager configuration. Important: To create multiple realms, ensure that your federated repository contains the required unique base entries. Stop and restart the appropriate servers for your installation environment, and then update the wkplc.properties file with the base entry information and rerun the wp-create-realm task. Repeat these steps until all realms are created.
5. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.
6. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading and then save your changes: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - realmName
   - realm.personAccountParent
   - realm.groupParent
   - realm.orgContainerParent
7. Run the ./ConfigEngine.sh wp-modify-realm-defaultparents -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to update the default parents per entity type and realm. Important: Stop and restart the appropriate servers for your installation environment before rerunning this task for any additional entity types and realms.
8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents*.

9. Optional: Perform the following steps to add additional base entries to the realm configuration; for example, if you had two additional base entries (base entry 1 and base entry 2) to add to the realm you just created, you would update the *wkplc.properties* file with the information from base entry 1 and then run this task. Then you would update the properties file with the information for base entry 2 and then run this task:
   A. Use a text editor to open the *wkplc.properties* file, located in the `wp_profile_root/ConfigEngine/properties` directory.
   B. Enter a value for the following required parameters in the *wkplc.properties* file under the VMM realm configuration heading:
      - **realmName**
      - **addBaseEntry**
   C. Save your changes to the *wkplc.properties* file.
   D. Run the `./ConfigEngine.sh wp-add-realm-baseentry -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to add additional LDAP base entries to the realm configuration.
   E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID; this step is required if you change the default realm:
    A. Create a new user in the *Manage Users and Groups* portlet to replace the current WebSphere Application Server administrative user.
    B. Create a new user in the *Manage Users and Groups* portlet to replace the current WebSphere Portal administrative user.
    C. Create a new group in the *Manage Users and Groups* portlet to replace the current group.
    D. Run the `./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task, from the `wp_profile_root/ConfigEngine` directory, to replace the old WebSphere Application Server administrative user ID and group ID with the new user and group. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.
    E. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.
    F. Run the `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task to replace the old WebSphere Portal administrative user ID and group ID with the new user and group. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.
    G. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

11. Optional: Perform the following steps to set the realm you created as the default realm. **Remember:** Only users defined in base entries that exist in the default realm are able to log into WebSphere Portal. If you find that a user cannot log in to WebSphere Portal, check to see if the base entry that contains the user exists in the default realm. You can run the `wp-query-realm-baseentry` task to see what base entries are part of the default realm. If the default realm is missing the base entry, run the `wp-add-realm-baseentry` task to add the base entry to the default realm.
A. Use a text editor to open the wkplc.properties file, located in the \texttt{wp_profile_root}/ConfigEngine/properties directory.

B. For \texttt{defaultRealmName}, type the \texttt{realmName} property value you want to use as the default realm.

C. Save your changes to the wkplc.properties file.

D. Run the \texttt{./ConfigEngine.sh wp-default-realm -DWasPassword=password} task, from the \texttt{wp_profile_root}/ConfigEngine directory, to set this realm as the default realm.

E. Stop and restart all necessary servers to propagate your changes.

12. Optional: Perform the following steps to query a realm for a list of its base entries:

A. Use a text editor to open the wkplc.properties file, located in the \texttt{wp_profile_root}/ConfigEngine/properties directory.

B. For \texttt{realmName}, type the name of the realm you want to query.

C. Save your changes to the wkplc.properties file.

D. Run the \texttt{./ConfigEngine.sh wp-query-realm-baseentry -DWasPassword=password} task, from the \texttt{wp_profile_root}/ConfigEngine directory, to list the base entries for a specific realm.

13. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

A. Use a text editor to open the wkplc.properties file, located in the \texttt{wp_profile_root}/ConfigEngine/properties directory.

B. Enter a value for \texttt{realmName} or leave blank to update the default realm.

C. Save your changes to the wkplc.properties file.

D. Run the \texttt{./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password} task, located in the \texttt{wp_profile_root}/ConfigEngine directory, to enable the distinguished name login.\textbf{Note:} After running this task to enable the full distinguished name login, you can run the \texttt{./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password} task to disable the feature.

E. Stop and restart all necessary servers to propagate your changes.

\textbf{Parent topic:} Configuring the default federated repository on Solaris
Adapting the attribute configuration on Solaris

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

After installation, IBM WebSphere Portal has a predefined set of attributes for users and groups. Your LDAP server may have a different set of predefined user and group attributes. To ensure proper communication between WebSphere Portal and your LDAP server, you can configure additional attributes and flag existing attributes as required or unsupported on a per repository basis or for all configure repositories.

LDAP servers can only handle attributes that are explicitly defined in their schema. The LDAP server’s schema is different from the WebSphere Portal schema but the two schemas should match for proper communication between WebSphere Portal and the LDAP server. The task to add the LDAP user registry does some basic attribute configurations depending on the type of LDAP server that you choose. You may, however, still need to adapt the WebSphere Portal configuration to match the LDAP schema; for example, if an attribute is defined in WebSphere Portal but not in the LDAP server, you will need to perform one of the following tasks to resolve this mismatch:
- Flag the attribute as unsupported for the LDAP server
- Introduce an attribute mapping that maps the WebSphere Portal attribute to an attribute defined in the LDAP schema

Perform the following tasks to adapt the attribute configuration to match the configured LDAP server(s) and your business needs:

1. **Querying the defined attributes on Solaris**
   After installing IBM WebSphere Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

2. **Adding attributes on Solaris**
   The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM WebSphere Portal and your user registry.

3. **Mapping attributes**
   After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

4. **Removing attributes**
   Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

**Parent topic:** Configuring WebSphere Portal to use a user registry on Solaris

**Previous topic:** Choosing your user registry model on Solaris

**Next topic:** Stand-alone server: Configuring WebSphere Portal to use dynamic groups on Solaris

**Related tasks**
Adding an LDAP user registry on Solaris
Adding an LDAP user registry over SSL on Solaris
Configuring a stand-alone LDAP user registry on Solaris
Configuring a stand-alone LDAP user registry over SSL on Solaris
Querying the defined attributes on Solaris

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

Warning: If you are using a database user registry or a property extension database, copy the database drivers to the AppServer_root/lib directory before executing this script.

Run the ./ConfigEngine.sh wp-query-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, any time during the configuration process or at runtime to query an overview of the currently defined attributes. This task creates the availableAttributes.html report, located in the wp_profile_root/ConfigEngine/log directory. The report contains one table that lists the available attributes for Users (PersonAccount) and one table that lists the available attributes for Groups. For each configured repository there is a column that indicates if the attribute is flagged as unsupported or if the attribute is mapped to a different LDAP attribute. Note: This task does not validate the existence of attributes in the LDAP schema.

Parent topic: Adapting the attribute configuration

Next topic: Adding attributes on Solaris
Adding attributes on Solaris

The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM® WebSphere® Portal and your user registry.

Perform the following steps to add new attributes to your user registry:

1. Install the required Enterprise Archive (.ear) file on WebSphere Application Server. Select the appropriate task to install the .ear file:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone environment</td>
<td>Perform the following steps to install the .ear file in a stand-alone environment: Open a command prompt. Navigate to the <code>wp_profile_root/ConfigEngine</code> directory. Run the <code>.ConfigEngine.sh wp-la-install-ear -DWasPassword=password</code> task.</td>
</tr>
<tr>
<td>Clustered environment</td>
<td>Perform the following steps to install the .ear file in a clustered environment: Open a command prompt. Navigate to the <code>wp_profile_root/ConfigEngine</code> directory on the primary node. Run the <code>.ConfigEngine.sh wp-la-install-ear -DWasPassword=dmgr_password -DServerName=dmgr -DNodeName=node_name</code> task. Where <code>node_name</code> is the name of the node where the deployment manager resides; you can find the <code>node_name</code> value in the WebSphere Application Server Administrative Console under System administrator &gt; Deployment Manager &gt; Runtime tab &gt; General Properties &gt; Node Name.</td>
</tr>
</tbody>
</table>

2. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

3. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

4. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Property Extension Properties heading:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>la.providerURL</td>
<td>See the properties file for specific information about the required parameters and for advanced parameters.</td>
</tr>
<tr>
<td>la.propertyName</td>
<td></td>
</tr>
<tr>
<td>la.entityTypes</td>
<td></td>
</tr>
<tr>
<td>la.dataType</td>
<td></td>
</tr>
<tr>
<td>la.multiValued</td>
<td></td>
</tr>
</tbody>
</table>

5. Save your changes to the `wkplc.properties` file.

6. Run the `.ConfigEngine.sh wp-add-property -DWasPassword=password` task to add the attribute to the user registry.

   Note: This task performs an EJB call to WebSphere Application Server, which must authenticate against WebSphere
Application Server. Depending on the configuration in the `sas.client.props` file, you may receive a popup window or a command line prompt asking for user identity and password. Enter the WebSphere Application Server user ID and password.

**Remember:** If you have multiple properties to add, repeat all steps, except for the `wp-la-install-ear` task, until all new attributes are added.

7. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents.*

---

**Related tasks**
- Adapting the attribute configuration
- Querying the defined attributes on Solaris
- Mapping attributes

---

**Parent topic:** Adapting the attribute configuration  
**Previous topic:** Querying the defined attributes on Solaris  
**Next topic:** Mapping attributes
Mapping attributes

After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

Perform the following steps to map attributes between WebSphere Portal and your LDAP server; if you have multiple LDAP servers, you will need to perform these steps for each LDAP server:

1. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

2. Enter a value for one of the following sets of parameters in the `wkplc.properties` file to identify your LDAP server:
   
   **Note:** Make sure you use the same values you used to configure your LDAP server.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>The following parameters are found under the LDAP attribute configuration heading: <strong>Note:</strong> See the properties file for specific information about the required parameters and for advanced parameters. <code>standalone.ldap.id</code> <code>standalone.ldap.host</code> <code>standalone.ldap.port</code> <code>standalone.ldap.sslEnabled</code> <code>standalone.ldap.bindDN</code> <code>standalone.ldap.bindPassword</code> <code>standalone.ldap.baseDN</code></td>
</tr>
<tr>
<td>Federated</td>
<td>The following parameters are found under the VMM Federated repository properties heading: <strong>Note:</strong> See the properties file for specific information about the required parameters and for advanced parameters. <code>federated.ldap.id</code> <code>federated.ldap.host</code> <code>federated.ldap.port</code> <code>federated.ldap.sslEnabled</code> <code>federated.ldap.bindDN</code> <code>federated.ldap.bindPassword</code> <code>federated.ldap.baseDN</code></td>
</tr>
</tbody>
</table>

3. Run one of the following tasks to check that all defined attributes are available in the configured LDAP user registry:

   **Table 2. Task to check that all defined attributes are available in the configured LDAP user registry.**

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td><code>./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password</code> task, from the <code>wp_profile_root/ConfigEngine</code> directory.</td>
</tr>
<tr>
<td>Federated</td>
<td><code>./ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password</code> task, from the <code>wp_profile_root/ConfigEngine</code> directory.</td>
</tr>
</tbody>
</table>

4. Open the `ConfigTrace.log` file, located in the `wp_profile_root/ConfigEngine/log` directory, to review the following output for the PersonAccount and Group entity type:
- The following attributes are defined in WebSphere Portal but not in the LDAP server
  - This list contains all attributes that are defined in WebSphere Portal but not available in the LDAP. Flag attributes that you do not plan to use in WebSphere Portal as unsupported. Map the attributes that you plan to use to the attributes that exist in the LDAP; you must also map the `uid`, `cn`, `firstName`, `sn`, `preferredLanguage`, and `ibm-primaryEmail` attributes if they are contained in the list.

- The following attributes are flagged as required in the LDAP server but not in WebSphere Portal
  - This list contains all attributes that are defined as "MUST" in the LDAP server but not as required in WebSphere Portal. You should flag these attributes as required within WebSphere Portal; see the step below about flagging an attribute as either unsupported or required.

- The following attributes have a different type in WebSphere Portal and in the LDAP server
  - This list contains all attributes that WebSphere Portal might ignore because the data type within WebSphere Portal and within the LDAP server do not match.

5. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

6. Enter a value for one of the following sets of parameters in the `wkplc.properties` file to correct any issues found in the config trace file: Table 3. Parameters to define in the `wkplc.properties` file to correct any issues found in the config trace file.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>The following parameters are found under the LDAP attribute configuration heading: <strong>Note:</strong> See the properties file for specific information about the required parameters and for advanced parameters. &lt;br&gt;<code>standalone.idstandalone.idap.items.nonSupportedstandalone.idap.attributes.nonSupported.deletestandalone.idap.attributes.nonSupported.deletestandalone.idap.attributes.mapping.idapNamestandalone.idap.attributes.mapping.poralNamestandalone.idap.attributes.mapping.poralNamestandalone.idap.attributes.mapping.entityTypes</code>&lt;br&gt;For example, the following values will flag <code>certificate</code> and <code>members</code> as unsupported attributes and will map <code>ibm-primaryEmail</code> to <code>mail</code> and <code>ibm-jobTitle</code> to <code>title</code> for both the <code>PersonAccount</code> and <code>Group</code> entity types.&lt;br&gt;<code>standalone.idap.attributes.nonSupported=certificate, members</code>&lt;br&gt;<code>standalone.idap.attributes.nonSupported.delete=standalone.idap.attributes.mapping.idapNamemail, title</code>&lt;br&gt;<code>standalone.idap.attributes.mapping.portalName=ibm-primaryEmail, ibm-jobTitle</code>&lt;br&gt;<code>standalone.idap.attributes.mapping.entityTypes=PersonAccount, Group</code></td>
</tr>
</tbody>
</table>
Save your changes to the `wkplc.properties` file.

Run one of the following tasks to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between WebSphere Portal and the LDAP user registry:

Table 4. Task to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between Portal and the LDAP user registry.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td><code>./ConfigEngine.sh wp-update-standalone-ldap-attribute-config -DWasPassword=password</code> task, from the <code>wp_profile_root/ConfigEngine</code> directory</td>
</tr>
<tr>
<td>Federated</td>
<td><code>./ConfigEngine.sh wp-update-federated-ldap-attribute-config -DWasPassword=password</code> task, from the <code>wp_profile_root/ConfigEngine</code> directory</td>
</tr>
</tbody>
</table>

Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents.*

Optional: Perform the following steps to flag an attribute as either unsupported or required for the entire WebSphere Portal environment instead of just for the specified LDAP:

A. Enter a value for the following required parameters in the `wkplc.properties` file: *Note:* See the properties file for specific information about the required parameters and for advanced parameters.

- `user.attributes.nonsupported`
- `user.attributes.required`

B. Save your changes to the `wkplc.properties` file.

C. Run the `./ConfigEngine.sh wp-update-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory.

D. Stop and restart all necessary servers to propagate your changes.

Parent topic: Adapting the attribute configuration

Previous topic: Adding attributes on Solaris
Next topic: Removing attributes

Related tasks
Starting and stopping servers, deployment managers, and node agents
Removing attributes

Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

Perform the following steps to remove an attribute from your database:

**Important:** Do not remove attributes that have already been populated with user values because this can cause database inconsistencies.

**Cluster note:** In a clustered environment, perform the following steps on the deployment manager and then resynch the nodes.

1. Perform the following steps to remove an attribute stored in a property extension database; this is an attribute that was added using the `wp-add-la-property` task.
   A. Open the tool you use to edit your database.
   B. Verify that your attribute name is available in the LAPROP table.
   C. Delete the required attributes from the LAPROP table.
   D. Open the `wimxmlextension.xml` file, located in the `wp_profile_root/config/cells/cellname/wim/model` directory.
   E. Locate and delete the `propertySchema` definition for the attributes that you deleted from the LAPROP table; for example:
   ```xml
   <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String"
   multiValued="true" propertyName="attribute_name">
   <wim:applicableEntityTypeName>PersonAccount</wim:applicableEntityTypeName>
   </wim:propertySchema>
   ```
   F. Save your changes to the `wimxmlextension.xml` file.
   G. Open the `wimconfig.xml` file, located in the `wp_profile_root/config/cells/cellname/wim/config` directory.
   H. Locate and delete the `propertiesNotSupported` definitions for the attributes that you deleted from the LAPROP table; for example:
   ```xml
   <config:propertiesNotSupported name="attribute_name">
   </config:propertiesNotSupported>
   ```
   I. Save your changes to the `wimconfig.xml` file.
   J. Stop and restart the server and WebSphere_Portal servers from the `wp_profile_root/bin` directory.

2. Perform the following steps to remove an attribute that is not stored in a property extension database:
   A. Open the `wimxmlextension.xml` file.
   B. Locate and delete the `propertySchema` definition for the attributes you previously added:
   ```xml
   <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String"
   multiValued="true" propertyName="attribute_name">
   <wim:applicableEntityTypeName>PersonAccount</wim:applicableEntityTypeName>
   </wim:propertySchema>
   ```
   C. Save your changes to the `wimxmlextension.xml` file.
   D. Open the `wimconfig.xml` file.
   E. Locate and delete the stanza that corresponds to the custom attribute you deleted from the `wimextension.xml` file; for example:
   ```xml
   <config:attributes name="attribute_name" propertyName="property_name">
   <config:entityTypes>PersonAccount</config:entityTypes>
   </config:attributes>
   ```
   F. Save your changes to the `wimconfig.xml` file.
G. Stop and restart the server1 and WebSphere_Portal servers.

**Parent topic:** Adapting the attribute configuration  
**Previous topic:** Mapping attributes
Stand-alone server: Configuring WebSphere Portal to use dynamic groups on Solaris

By default, WebSphere Portal is enabled for static groups. However, the Virtual Member Manager (VMM) allows users to be members of either static or dynamic groups. Static groups are those where a persistent binding exists between a group and its members. Dynamic groups are those where a search query is defined to retrieve the members of a group. If you have your LDAP server configured to use dynamic groups, complete the steps in this task for WebSphere Portal to use dynamic group queries when you setup your LDAP server.

Perform the required tasks to configure either a stand-alone or federated LDAP server security.

The steps in this task use **groupofURLs** as the object class for dynamic groups and **memberURL** as the dynamic membership attribute. The actual values for object classes and dynamic membership attributes can vary depending on your LDAP server. For this reason, you should export an LDIF file to verify the object classes and dynamic membership attributes. Either refer to your LDAP documentation or ask your LDAP administrator for instructions on exporting an LDIF file.

**Clustered environments:** Perform the following steps on the Deployment Manager then synchronize the nodes.

To configure WebSphere Portal to use dynamic groups, do the following:

1. Choose the appropriate set of steps, depending on your LDAP server environment: Table 1. Steps for enabling dynamic groups

<table>
<thead>
<tr>
<th>LDAP server environment</th>
<th>Steps to perform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone LDAP server or federated LDAP server(s)</td>
<td>Navigate to the following directory: <code>wp_profile_root/cells/cell_name/wim/config</code>. Locate and open <code>wimconfig.xml</code> with any text editor. Add the following line to the <code>&lt;config:groupConfiguration&gt;</code> tag: <code>&lt;config:dynamicMemberAttributes name=&quot;memberurl&quot; objectClass=&quot;groupofurls&quot;/&gt;</code>. Save and close <code>wimconfig.xml</code>.</td>
</tr>
<tr>
<td>Federated LDAP server(s)</td>
<td>Log in to the administration console. Select <code>Security &gt; Secure administration, applications, and infrastructure</code>. Under <code>Available realm definitions</code>, select <code>Federated repositories</code> and click <code>Configure</code>. Under <code>Related Items</code>, click <code>Manage repositories</code>. Select the appropriate repository from the list. Under <code>Additional Properties</code>, click <code>Group attribute definition</code> then click <code>Dynamic member attributes</code>. Click <code>New</code> and specify values for the <code>Name</code> and <code>Object class</code> fields as appropriate. For example, <code>Name: memberurl Object class: groupofurls</code>. Click <code>OK</code> and save the changes to the master configuration.</td>
</tr>
</tbody>
</table>

2. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under `Related tasks: Starting and stopping servers, deployment managers, and node agents.`
Stand-alone server: Enabling referrals for your LDAP user registry on Solaris

Referrals redirect object requests from one LDAP server to another when objects do not exist or cannot be located in a particular directory tree. You should enable referrals if your environment has more than one user registry existing on multiple servers or domains.

To configure your portal to use LDAP referrals, do the following:

1. Use any text editor to open the wkplc.properties file in the following directory: wp_profile_root/ConfigEngine/properties.

2. Specify values for the following parameters:
   - `et.ldap.id` = ID_of_your_LDAP_server
   - `et.ldap.host` = hostname_of_your_LDAP_server
   - `et.ldap.referral` = follow

3. Save and close wkplc.properties.

4. Run the following task from the wp_profile_root/ConfigEngine directory to create an LDAP entity type:
   - UNIX: ./ConfigEngine.sh wp-update-et-ldap -DWasPassword=password
   - Windows: ConfigEngine.bat wp-update-et-ldap -DWasPassword=password
   - IBM® i5/OS: ConfigEngine.sh wp-update-et-ldap -DWasPassword=password

5. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

Parent topic: Configuring WebSphere Portal to use a user registry on Solaris

Previous topic: Stand-alone server: Configuring WebSphere Portal to use dynamic groups on Solaris
Tune your servers

Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

- **Prerequisites**
  - Configuring databases
  - Base Portal Tuning Scenarios

Tuning a WebSphere Portal environment involves tuning and configuring the various systems and components of the environment. The tuning guide provides general concepts and details specifics configuration instructions. Instructions are included for:
- Configuring the application server and the resources defined for that application server
- Determining the cloning strategy for expanding or extending the environment
- Tuning the database(s) and database server
- Tuning the directory server and its database
- Tuning the Web server
- Tuning the operating system and network
- Tuning the WebSphere Portal services

**Parent topic:** Setting up a stand-alone server on Solaris

**Previous topic:** Configuring WebSphere Portal to use a user registry on Solaris

**Related Information**
- WebSphere Portal and Lotus Web Content Management 6.1.x Performance Tuning Guide
- IBM WebSphere Portal Performance Troubleshooting Guide
Setting up a stand-alone server on Windows

After you have prepared the operating system you are ready to install WebSphere Portal.

- Prerequisites
  - [Technotes for installation and configuration issues](#)

Select the operating system on which you are installing WebSphere Portal.

1. Preparing your Windows operating system
   View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

2. Installing WebSphere Portal on Windows
   IBM WebSphere Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

3. Setting up a database
   Before transferring default data to the production database server, you need to prepare the database server. Database preparation includes creating user IDs and databases. For this installation path, stand-alone production server, remote databases servers are used.

4. Preparing a remote Web server when portal is installed on Windows
   Install and configure the Web server plug-in, provided by IBM WebSphere Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

5. Configuring WebSphere Portal to use a user registry on Windows
   Configure user registry security on IBM WebSphere Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

6. Tune your servers
   Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

Parent topic: Setting up a stand-alone production server
Preparing your Windows operating system

View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

Perform the following steps to prepare your operating system:

1. Check that the system logon user ID you will use during installation has the following permissions and rights:
   - The user ID must already exist prior to installation.
   - The user ID must belong to the Administrators group.

2. Perform the following steps to determine if a user account is a member of the Administrators group:
   A. Click Start > Programs > Administrative Tools > Computer Management.
   B. Expand Local Users and Groups and select Groups.
   C. Open the Administrators group to see what members belong to it.
   D. Add the user to the Administrators group if necessary.

3. Consider the following recommendations when installing to avoid excessively long path names:
   Note: If you exceed the 259 maximum character length, you may receive one of the following error messages during configuration or in the wpinstalllog.txt file:
   - The input line is too long.
   - The syntax of the command is incorrect.
   - The filename is too long.

   A. Use a short installation path. For example, use C:\WebSphere instead of C:\Program Files\IBM\WebSphere.
   B. Specify node names; do not use names longer than 5 characters. For example, you might use node1 instead of longnodename01.
   C. Name WAR files with less than 21 characters. If necessary, modify the file name before installing.

Parent topic: Setting up a stand-alone server on Windows
Next topic: Installing WebSphere Portal on Windows
Installing WebSphere Portal on Windows

IBM® WebSphere® Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

Review the Supported hardware and software requirements for your operating system and ensure that all hardware and software matches the minimum product level before installing WebSphere Portal. Review Installation methods, options, and sources before installing to determine your installation source location and installation method.

Note: If you plan to use IBM WebSphere Application Server Version 7.0, you must purchase and install it before installing WebSphere Portal on the existing WebSphere Application Server Version 7.0 instance.

Perform the following steps to install WebSphere Portal:

1. Type ping yourserver.yourcompany.com on a command line to verify that your fully qualified host name is properly configured.
2. Type ping localhost on a command line to verify that your network is properly configured.
3. If you are installing with an existing WebSphere Application Server instance, ensure it is installed at the supported level and has the following features installed:
   
   **Restriction:** Installing WebSphere Portal Version 6.1.0.1 or higher on an existing WebSphere Application Server Version 7.0 instance where WebSphere Portal Version 7.0 was installed is unsupported.
   
   **Important:** Besides ensuring that the existing WebSphere Application Server instance is at the supported level, you will also need to ensure that Apache Derby is installed at the supported level before installing WebSphere Portal. See WebSphere Portal detailed system requirements for supported levels.

4. If you are installing on a server with a firewall, antivirus, and/or desktop search engine enabled, disable them before installing. If you do not disable them and the installation program detects them, a warning message displays during the installation.

5. Choose one of the following installation commands based on the installation method you want to use to install WebSphere Portal; see Installation Methods for more information:

   **Advance installation parameters:** To customize your installation, you can add various parameters to your installation commands; for example, you can specify your port information. See Advanced installation parameters for information.

   **Restriction:** When defining your installation path, the ONLY supported characters are ASCII letters [A-Z and a-z], numbers [0-9], slashes [/ or \, depending on your operating system], and the dash [-].

   **Table 1. Installation task options**

<table>
<thead>
<tr>
<th>Installation type</th>
<th>Task</th>
</tr>
</thead>
</table>

958
**Graphical user interface**
install.bat

**Console mode**
install.bat -console from the PortalExpress subdirectory

**Silent Install**
install.bat -options "path_to_file\response_filename", where path_to_file is the full path to the response file and response_filename is the name of the file. A sample install response file (installresponse.txt) and a sample uninstall response file (uninstallresponse.txt) are located in the setup CD root directory. **Important:** Do not place the response file in a path that contains a space and do not put a space in the file name.

**Note:** If the installation program does not detect a WebSphere Application Server instance that you know exists, exit the installation program and pass the WebSphere Application Server instance location using the command line; for example, install.bat -W was.undetectedWas="/my/WAS/location".

**Note:** If the GUI or console mode installation program fails to detect ports for either WebSphere Application Server or WebSphere Portal, a warning message displays and the installer offers another chance to enter the values. If the silent installation fails to detect ports for either WebSphere Application Server or WebSphere Portal, the installer will exit.

6. **Verify your installation was successful:** access WebSphere Portal using the `http://yourserver:yourport/wps/portal` format. Run the `ConfigEngine.bat list-server-ports -DWasPassword=password` task from the `wp_profile_root\ConfigEngine` directory to generate the `wp_profile_root\ConfigEngine\log\wp_PortMatrix.txt` file that lists the WebSphere Portal ports for your installation.

7. **Optional:** After the WebSphere Portal installation completes successfully, run the `ConfigEngine.bat configure-express -DPortalAdminPwd=password -DWasPassword=password` task, from the `wp_profile_root\ConfigEngine` directory, if you want to enable the sample IBM Lotus Web Content Management content, such as internet and intranet sample sites.

   **Restriction:** Run the `configure-express` task before configuring your database, user registry, context root, security, etc. If you ran any tasks other than the install task, do not run this task.

   The sample content includes: **Note:** See [http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_ovw.html](http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_ovw.html) for tutorials on how to use the sample content.

- Creates a group called `contentAuthors`; members of this group are given privileges to create content in the sample Internet and intranet sites. Navigate to the **Administration** area and then click **Access > Users and Groups**.
- Creates two new Web Content Management Libraries: "Internet Web Content 6.1.0" and "Intranet Web Content 6.1.0". Navigate to the **Administration** area and then click **Portal Content > Web Content Libraries**.
- Adds a portlet filter and applies the filter to various portlets in the sample Internet and intranet sites. You can see the definition of the filter in the WebSphere Application Server Administration console and examining the custom resources under the **Environment** area.
- Creates two new theme policies: InternetStyle and IntranetStyle. These styles are applied to sample Internet and intranet sites. Navigate to **Theme Customizer** and then select the style.
- Creates several portlet clones of the Web Content Management rendering portlet. These portlet clones are used on sample Internet and intranet sites.
- Creates several sample credential slots, including "Default slot for E-mail", "Default slot for Feeds", "Default slot for Miscellaneous", "Default slot for Web Clipping", and "Default slot for Web Content Management". Navigate to the
Administration area and then click **Access > Credential Vault > Manage System Vault Slots.**
- Modifies the portlet palette to include some new portlets. Click the **Portlets** link to see the portlet palette.

8. Optional: If you ran the **configure-express** task, the owner of the items in the Web content libraries containing the Internet and Intranet Site Template content will be listed as `uid=xyzadmin,o=defaultWIMFileBasedRealm`. To update the owner information for these items to correspond to your portal administrator ID, complete the following steps:

A. Edit the `wp_profile_root\PortalServer\wcm\shared\app\config\wcm\services\MemberFixerModule.properties` file.

B. Add the following line to the file:

```
uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
```

Replace `portal_admin_DN` with the distinguished name of the portal administrator.

C. Save your changes and close the file.

D. Run the `ConfigEngine.bat action-express-memberfixer -DPortalAdminPwd=***password*** -DWasPassword=***password***` task, located in the `wp_profile_root\ConfigEngine` directory.

9. Optional: Perform the following steps if you need to change the ports for WebSphere Application Server or WebSphere Portal:

**Note:** The starting port parameter is required for a successful completion of the **modify-ports-by-startport** task. Once you specify a start port, this port becomes the base for assigning port values. The code increments this value as each port is assigned, which means that the WebSphere Portal ports will be assigned incrementally starting with the port defined with the **modify-ports-by-startport** task.

A. Stop the `server1` and WebSphere_Portal servers.

B. Run one of the following commands for each server you need to change:

<table>
<thead>
<tr>
<th>Method</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting port number</td>
<td><code>ConfigEngine.bat modify-ports-by-startport -DWasPassword=***password*** -DModifyPortsServer=servername -DStartPort=starting port number</code></td>
</tr>
<tr>
<td>Port file<strong>Note:</strong> Sample port files are available on the Setup disc</td>
<td><code>ConfigEngine.bat modify-ports-by-portsfile -DWasPassword=***password*** -DModifyPortsServer=servername -DPortsFile=full path to ports file</code></td>
</tr>
</tbody>
</table>

The following is an example of the information within a port file although the port values will be different based on your environment:

```
BOOTSTRAP_ADDRESS=10031
SOAP_CONNECTOR_ADDRESS=10033
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=10032
CSV2_SSL_SERVERAUTH_LISTENER_ADDRESS=10025
CSV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=10036
WC_adminhost=10027
WC_defaulthost=33344
DCS_UNICAST_ADDRESS=10029
WC_adminhost_secure=10039
WC_defaulthost_secure=10035
SIB_ENDPOINT_ADDRESS=10026
SIB_ENDPOINT_SECURE_ADDRESS=10037
SIB_MQ_ENDPOINT_ADDRESS=10030
SIB_MQ_ENDPOINT_SECURE_ADDRESS=10028
ORB_LISTENER_ADDRESS=10034
```

C. Restart the server1 and WebSphere_Portal servers.

**Parent topic:** Setting up a stand-alone server on Windows

**Previous topic:** Preparing your Windows operating system

**Next topic:** Setting up a database
Related concepts
Installation methods
IBM Support Assistant Lite for WebSphere Portal

Related tasks
Creating and maintaining multiple profiles on Windows
Using the member fixer tool with IBM Lotus Web Content Management

Related reference
Advanced installation parameters
Setting up a database

Before transferring default data to the production database server, you need to prepare the database server. Database preparation includes creating user IDs and databases. For this installation path, stand-alone production server, remote databases servers are used.

Select the database server that are using.

- **Configuring a local database**
  The database can be installed on the same server as WebSphere Portal. Having the database on the same server has advantages and disadvantages. This is a common scenario for a proof-of-concept installation where you want to test out the capabilities and features. Another scenario that may use this configuration is a stand-alone server installation.
  Installing the database on another server has the advantage of distributing the processing and workload.

- **Setting up a remote DB2 database**
  You can install DB2 on the same server as WebSphere Portal, also referred to as local, or on a different server, referred to as remote. If you plan to use a local DB2 database, scripts are provided to simplify the database creation and transfer.
  More information about the local database is provide in the single server installation instructions. For a remote database you must complete the transfer manually.

- **Setting up a remote Oracle database on Windows for a stand-alone production server**
  Setting up the Oracle database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

- **Setting up a remote Oracle RAC database on Windows for a stand alone production server**
  Setting up the Oracle RAC database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

- **Setting up a remote SQL Server database on Windows for a stand-alone production server**
  To setup a remote Microsoft SQL Server Enterprise Edition database, create user IDs and databases on a remote server. Tasks are provided to assist with creating the user IDs and the databases. Before you can use the tasks, you must modify properties files.

- **Verifying databases**
  After you configure IBM WebSphere Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the **SystemOut.log** and **SystemErr.log** files.

**Parent topic:** Setting up a stand-alone server on Windows

**Previous topic:** Installing WebSphere Portal on Windows

**Next topic:** Preparing a remote Web server when portal is installed on Windows
Configuring a local database

The database can be installed on the same server as WebSphere Portal. Having the database on the same server has advantages and disadvantages. This is a common scenario for a proof-of-concept installation where you want to test out the capabilities and features. Another scenario that may use this configuration is a stand-alone server installation. Installing the database on another server has the advantage of distributing the processing and workload.

- Installing DB2 on the same server as WebSphere Portal
  View information on installing DB2 for use with WebSphere Portal.

- Configuring JCR collation support
  View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

- Modifying database properties
  Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

- Setting up databases
  This section provides information on using ConfigEngine tasks to create databases and users.

- Assigning custom table spaces
  The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

- Configuring WebSphere Portal to use DB2
  View information on manually transferring data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

- Configuring DB2 for large file handling in Web Content Management
  If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the fullyMaterializeLobData property in the WebSphere Application Server administrative console.

-Verifying databases
  After you configure IBM WebSphere® Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.

Parent topic: Setting up a database
Installing DB2 on the same server as WebSphere Portal

View information on installing DB2 for use with WebSphere Portal.

1. Before installing DB2, log in with a user ID that has administrative authority. This user should have the following specifications:
   - Belong to the local Administrator group
   - Act as part of the operating system
   - Have permissions to create a token object
   - Windows 2003 only: Have permissions to adjust memory quotas for a process
   - Have permissions to replace a process level token

   To edit user rights:
   - For the first two specifications: Click Start > Programs > Administrative Tools > Computer Management > Local Users and Groups.
   - For the last four specifications: Click Start > Programs > Administrative Tools > Local Security Policy. Then, click Local Policies > User Rights Assignment.

2. To install DB2, follow the instructions that are provided with the DB2 documentation to install DB2 and the required fix pack.

Parent topic: Configuring a local database
Configuring JCR collation support

View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

1. Stop the WebSphere Portal server:
   ```bash
   stopServer.bat WebSphere_Portal -username admin_userid -password admin_password
   ```
2. If using a remote DB2 server, copy the following files from the WebSphere Portal server to a temporary directory on the remote DB2 server:
   ```bash
   PortalServer_root/jcr/prereq.jcr/config/collation.jar
   PortalServer_root/jcr/prereq.jcr/config/registerCollationUDFTemplate.sql
   ```
3. Set up collation on the database where the JCR domain is located. Log in to the database machine with a userid that is authorized and configured to use the appropriate DB2 instance. For example, a common userid is `db2inst1`.
   A. Change to the directory `db2home/function`.
   B. Execute the command:
      ```bash
      Remote DB2: db2home/java/jdk/bin/jar -xvf temporary location/collation.jar icm/CollationUDF.class
      Local DB2: db2home/java/jdk/bin/jar -xvf PortalServer_root/jcr/prereq.jcr/config/collation.jar icm/CollationUDF.class
      ```
   C. Change to the directory `wp_profile_root\PortalServer\jcr\config`. If using a remote DB2, change to the temporary directory where you copied the files in a previous step.
   D. Open the file `registerCollationUDFTemplate.sql` and change all `SCHEMA` references to the JCR schema; for example, `JCR`. The value set for `SCHEMA` should match the value set for the `jcr.DbSchema` property.
   E. Connect to the JCR database by running `db2 connect to <jcrdb> user <userid> using <password>`. Execute the script by running the command.
      ```bash
      Remote DB2: db2 -tvf temporary location/registerCollationUDFTemplate.sql
      Local DB2: db2 -tvf wp_profile_root\PortalServer\jcr\config\registerCollationUDFTemplate.sql
      ```
   F. Disconnect from the JCR database.
   G. Restart the DB2 instance.

4. Verify that the UDF is registered properly.
   A. Log in as the `db2instanceID`. Open a DB2 terminal window, and type `db2`. From the command line, type the command:
      ```bash
      connect to JCRDB user userid using password
      ```
   B. Execute the following command in the previous substep. If the command completes successfully, the UDF is registered correctly as follows:
      ```bash
      values schema.sortkeyj('abc','en')
      ```

5. Edit the `icm.properties` file, located in `wp_profile_root\PortalServer\jcr\lib\com\ibm\icm` directory. Add the following section to the end of the file:
   ```properties
   # Enable/Disable collation support for all DB2 platforms
   jcr.query.collation.db2.enabled = true
   
   # Database specific collation mappings
   # These mappings apply map a Java locale name into a collation name
   ```
# supported by the underlying database.
# Example mappings for DB2 platform

# English
jcr.query.collation.en = en

# Swedish
jcr.query.collation.sv = sv

jcr.query.collation.zh = zh
jcr.query.collation.de = de
jcr.query.collation.da = da
jcr.query.collation.hu = hu
jcr.query.collation.jp = jp

6. Enter the following command to start the WebSphere Portal server: `startServer.bat WebSphere_Portal`

Parent topic: Configuring a local database
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

Tip: For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Working with properties files:
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:

```
release.DbName = release
jcr.DbName = jcrdb
feedback.DbName = fdbkdb
likeminds.DbName = lmdb
community.DbName = commdb
customization.DbName = custdb
```
- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:

```
release
customization
community
jcr
feedback
likeminds
```
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:

```
dbdomain.DbType
dbdomain.DbName
dbdomain.DbUrl
```
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment. Note: The wkplc_comp.properties file by default has the properties populated for the Apache Derby database. The source.dbdomain.properties represent the properties for the source database. These values should not be modified unless you are transferring from a supported database other than Apache Derby.
   A. For dbdomainDbType, type db2.
   B. For dbdomainDbName, type the name of the WebSphere Portal domain database. Note: This value is also the database element in the dbdomainDbName property.
   C. For dbdomainDbName, type the name of the WebSphere Portal domain database. DB2 database names cannot exceed eight (8) characters. Note: This value is also the database element in the dbdomainDbName property.
   D. For dbdomainDbSchema, type the schema name of the database domain. Note: Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   E. For dbdomainDataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   F. For dbdomainDbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. Note: The database element of this value should match the value of DbName.
   G. For dbdomainDbUser, type the user ID for the database administrator.
   H. For dbdomainDbPassword, type the password for the database administrator.
   I. Optional: For dbdomainDbRuntimeUser, type the user ID of the database user that should be used by WebSphere Portal to connect to the database at runtime.
   J. If dbdomainDbRuntimeUser is specified, you must set dbdomainDbRuntimePassword to be the password of the runtime database user.
   K. For dbdomainDBA_DbUser, type the database administrator user ID for privileged access operations during creation of the database.
   L. For dbdomainDBA_DbPassword, type the database administrator password for privileged access operations during creation of the database.
3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties.

**Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

- **A.** For `source.domainDbType`, type of the database you are currently configured to use. The value for `source.domainDbType` is Derby by default.
- **B.** For `source.domainDbName`, type the name of the database domain you are currently using.
- **C.** For `source.domainDbSchema`, type current schema identifier for objects within the database for this domain.
- **D.** For `source.domainDataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.
- **E.** For `source.domainDbUrl`, type the url currently used to access your database.
- **F.** For `source.domainDbUser`, type the name of the user accessing this database.
- **G.** For `source.domainDbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

- **A.** For `db2.DbDriver`, type the name of the JDBC driver class.
- **B.** For `db2.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.
- **C.** For `db2.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

- **A.** For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Configuring a local database
Setting up databases
This section provides information on using ConfigEngine tasks to create databases and users.
Before you begin, ensure that the following prerequisites are met:
- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

The following steps are the same for root and non-root users, except that the create-database task cannot be run by a non-root user.
1. Change to the directory `wp_profile_root\ConfigEngine`
2. To create the databases, type the following command:
   ```
   ConfigEngine.bat create-database -DWasPassword=password
   ```
3. Check the services file on the DB2 server system. If it does not specify DB2 connection and interrupt service ports, specify the ports for your operating system.
   A. Use a text editor to open the file `%SYSTEMROOT%\system32\drivers\etc\services`.
   B. Add the text `db2c_db2 50000/tcp`, where `db2` is the default instance. **Note:** Ensure the port number used is not already in use. If 50000 is already in use, select a different port number.
4. To create the database users, type the following command:
   ```
   ConfigEngine.bat setup-database -DWasPassword=password
   ```
   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:
   ```
   '<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql'.
   ```

Parent topic: Configuring a local database
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties` file.
- The page size of table spaces used by WebSphere Portal must be 8192 bytes.
- For details on creating table spaces refer to the documentation for the database.

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties` that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`
   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
     - `release`
     - `community`
     - `customization`
     - `jcr`
     - `feedback`
     - `likeminds`
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword `IN` and a space. For example: `community.COMP_INST.tablespace=IN COMM8KSPACE`
   Repeat this step for each domain that you are transferring.
4. Save and close `dbdomain.space_mapping.properties`.
5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - Windows: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
   - UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
   - i5/OS: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

Parent topic: Configuring a local database
Configuring WebSphere Portal to use DB2

View information on manually transferring data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

**Tips:**

- To run these tasks as a non-root user, you must first run the task `chown -R non-root_userWebSphereDir`.

- If you are transferring from Oracle, the `open_cursors` setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

- Be sure that DB2 is started by checking the service. If attempts to restart result in a logon failure message, then go to the DB2 properties and reenter the password.

**Steps for transferring data to another supported database**

1. If you are running a type 2 connection, before transferring data edit the `db2cli.ini` file. Failure to follow these steps will cause the database transfer to hang at the task `action-process-constraints`.

   A. Locate the file `C:\Program Files\IBM\SQLLIB\db2cli.ini`.

   B. Add the following to the end of the file. Leave an empty line after `ReturnAliases=0`.

      ```
      [COMMON]
      DYNAMIC=1
      ReturnAliases=0
      ```

2. Open a command prompt and change to the directory `wp_profile_root\ConfigEngine`.

3. Enter the following commands to validate configuration properties.

   ```
   ConfigEngine.bat validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ConfigEngine.bat validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```

4. From the same command prompt as the previous steps, change to the directory `wp_profile_root\bin`.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td><code>stopServer.bat server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td><code>stopServer.bat WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>
6. Transfer the database. **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.
   
   A. Change to the directory `wp_profile_root\ConfigEngine`.
   
   B. Enter the following command: `ConfigEngine.bat database-transfer -DWasPassword=password`

   **Note:**
   - To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command: `ConfigEngine.bat database-transfer -DTransferDomainList=jcr -DWasPassword=password`
   
   - If you have been storing data in IBM® Derby for a long time, database transfer could fail with OutOfMemory exceptions. If database transfer fails, add the following property to the command in this step: `ConfigEngine.bat database-transfer -DDbtJavaMaxMemory=1536M -DWasPassword=password`

   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

   If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

7. After transferring the database tables, perform a reorg check to improve performance. Perform this step for each database alias in the property file.
   
   A. Connect to a database with the following command: `db2 connect to database_alias user db2admin_userid using password`

   **Note:** Additional options might be required if additional security has been installed. Refer to DB2 Universal Database commands by example for links to the command reference.

   B. After it is connected, run the following command from the DB2 prompt: `db2 reorgchk update statistics on table all > xyz.out`

   C. Look in the reorg column for entries marked with a * (star or asterisk) in the file `xyz.out`. For each line with a *, note the `tablename` and run the following command for each `tablename`: `db2 reorg table tablename db2 terminate db2rbind database_name -l db2rbind.out -u db2_admin -p password`

   D. The output file `db2rbind.out` is only created when there is an error for the `db2rbind` command.

8. Change to the directory `wp_profile_root\bin`.

9. Enter the following command to start the WebSphere Portal server: `startServer.bat WebSphere_Portal`
Configuring DB2 for large file handling in Web Content Management

If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

**Note:** You only need to perform these steps if you are using Web Content Management.

1. Log into the WebSphere Application Server administrative console.
2. Click **Resources > JDBC > Data sources.**
3. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
4. Select the name of the data source that is defined in `wkplc_comp.properties` for the JCR database domain. The default data source is `wpdbDS`.
5. Click **Custom properties.**
6. Ensure that the `fullyMaterializeLobData` property is set to `false`.

**Parent topic:** Configuring a local database
Verifying databases
After you configure IBM® WebSphere® Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the `SystemOut.log` and `SystemErr.log` files.

You can verify the database connection using IBM WebSphere Application Server or by opening WebSphere Portal in a browser.

- To verify that the WebSphere Portal application server is running by using WebSphere Application Server, complete these steps:

  1. Open the WebSphere Application Server administrative console by entering the following address in a browser:

     http://hostname.example.com:10027/ibm/console

     where `hostname.example.com` is the fully qualified host name of the machine where WebSphere Portal is running and 10027 is the default transport port that is created by WebSphere Application Server.

  2. Log into the administrative console.

  3. Depending on your version of WebSphere Application Server, click the appropriate option:

     - For **WebSphere Application Server Version 6.1**: Click **Resources & JDBC Providers**.
     - For **WebSphere Application Server Version 7.0**: Click **Resources > JDBC > JDBC Providers**

  4. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.

  5. Select the name of the data source that is defined in `wkplc_comp.properties`. The default data source is `wpdbDS`.

  6. Select the name of the JDBC provider that is specified in `wkplc_dbtype.properties`. The default JDBC provider is `wpdbJDBC_dbtype`, where `dbtype` is replaced by the value that matches your environment.

  7. Click **Test Connection** to verify the database connection. If configuration parameters have been changed, you might need to restart WebSphere Application Server for the test to complete.

- To verify that the WebSphere Portal application server is running by opening WebSphere Portal in a browser, enter the following URL in a supported browser:

  http://hostname.example.com:10040/wps/portal

  where `hostname.example.com` is the fully qualified host name of the machine where WebSphere Portal is running and 10040 is the default transport port that is created by WebSphere Application Server.

Parent topic: Configuring a local database
Setting up a remote DB2 database

You can install DB2 on the same server as WebSphere Portal, also referred to as local, or on a different server, referred to as remote. If you plan to use a local DB2 database, scripts are provided to simplify the database creation and transfer. More information about the local database is provide in the single server installation instructions. For a remote database you must complete the transfer manually.

1. Installing DB2
   View information on installing DB2 for use with WebSphere Portal.

2. Create users
   View the steps to set up users for DB2 to work with WebSphere Portal.

3. Creating remote databases
   A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

4. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

5. Modifying database properties
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

6. Optional: Configuring JCR collation support
   View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

7. Configuring WebSphere Portal to use DB2
   View information on manually transferring data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

8. Configuring DB2 for large file handling in Web Content Management
   If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the fullyMaterializeLobData property in the WebSphere Application Server administrative console.

9. Optional: Changing driver types
   WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Parent topic: Setting up a database
Installing DB2

View information on installing DB2 for use with WebSphere Portal.

Before you begin:

- **(For Type 2 drivers only):** DB2 Connect must be installed on the same system as WebSphere Portal, and have the same name as the server profile name.
- When you install DB2 using the DB2 installation program, it automatically creates a DB2 administrative user with the correct operating system rights.

1. If you are using the JDBC driver in type 2 mode, configure your DB2 Connect client with the following commands. If you are using a remote database, complete this step separately from the WebSphere Portal installation.
   - `db2 update dbm cfg using tp_mon_name WAS`
   - `db2 update dbm cfg using spm_name hostname, where hostname is the host name of WebSphere Portal.`
     
     Because the default for spm_name is the hostname itself, specifying the hostname parameter is optional. If your hostname is more than eight characters, use empty double quotes (" "). For example, `db2 update dbm cfg using spm_name " "`.

2. Before installing DB2, log in with a user ID that has administrative authority. This user should have the following specifications:
   - Belong to the local Administrator group
   - Act as part of the operating system
   - Have permissions to create a token object
   - Windows 2003 only: Have permissions to adjust memory quotas for a process
   - Have permissions to replace a process level token

   To edit user rights:
   - For the first two specifications: Click **Start > Programs > Administrative Tools > Computer Management > Local Users and Groups**.
   - For the last four specifications: Click **Start > Programs > Administrative Tools > Local Security Policy.** Then, click **Local Policies > User Rights Assignment**.

3. To install DB2, follow the instructions that are provided with the DB2 documentation to install DB2 and the required fix pack.

**Parent topic:** Setting up a remote DB2 database

**Next topic:** Create users
Create users

View the steps to set up users for DB2 to work with WebSphere Portal.

Before you begin: You should have completed Installing DB2.

- **Prerequisites**
  - Installing DB2

For all databases, use one user with administrative rights on the operating system and the DB2 installation. This user can be the database administrative user that is created automatically by the DB2 installation program. This user is the database configuration user that will be used for configuration tasks: creating database tables and performing database transfer. If you choose to have WebSphere Portal also create the databases, the database user should be given SYSADM rights. You only need to manually create an administrator ID when you do not want to use an existing DB2 administrator ID.

A common user name is `db2inst1`, but you can assign any user name as long as it has administrative access and follows the limitations listed here. Do not change the user name after creating it.

The limitations on user names are:

- User names can contain one to 20 characters.
- Group and instance names can contain one to eight characters.
- Names cannot be any of the following: `users` `admins` `guests` `public` `local`
- Names cannot begin with: `IBM` `SQL` `SYS`
- Names cannot include accented characters.
- Create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Following are the basic steps to create new users and grant privileges. For more information, refer to the DB2 product documentation.

1. From the DB2 Control Center, expand the object tree until you find the DB Users folder.
2. Right-click the DB Users folder and select **Create/Add**. The Create DB User notebook opens.
3. In the User name field, type a user ID.
4. Assign an authority for the user.
5. Click **OK**.

**Parent topic:** Setting up a remote DB2 database  
**Previous topic:** Installing DB2  
**Next topic:** Creating remote databases
Creating remote databases

A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

Before creating these databases, note the following information:

- Use IBM® DB2 Universal Database™ Enterprise Server Edition JDBC driver, JCC, type 4.
- If the DB2 JDBC type 4 driver is used, only the instructions on the remote server are required. You do not need to install the DB2 client software, and you do not need to complete the steps related to the DB2 client.
- The client software, DB2 Connect, must be correctly configured to connect to the remote DB2 server instance, for example, db2inst1.
- These instructions assume that a remote DB2 server and DB2 Connect are already installed and running. For information on installing DB2 Connect, see the DB2 product documentation.

**Prerequisites**

- **Installing DB2**
- **Create users**

To create a database, you must be a DB2 System Administrator with sufficient database privileges (SYSADM or at a minimum SYSCTRL).

1. Log in to the DB2 server system as a user with sufficient database privileges.
2. Ensure that the database user has been created, granted appropriate privileges, and has a password assigned to it. If the user has not been created, refer to the Creating users topic for information on how to create users.
3. Initialize a DB2 command environment by opening a command prompt and typing `db2cmd`. In this mode, you must type `db2` at the beginning of each command and use double quotation marks (""") around the entire command. The following example shows how to use the double quotes; it is not an actual command that you run:

`db2 "CREATE REGULAR TABLESPACE SMS PAGESIZE 4 K MANAGED BY SYSTEM USING ('sms') BUFFERPOOL SMSPOOL"`

**Note:** If you are using the Command Line Processor (CLP), refer to your DB2 documentation for details. The command prompt is `db2=>`. In this mode, commands can be entered without the `db2` prefix or the double quotation marks. However, the following steps assume you are not using the CLP and are entering commands from the $ prompt.

4. Run the following commands on the DB2 server system to configure the DB2 database instance:

5. **(Important)** Failure to complete this step will result in an unsuccessful database transfer. Run the following commands on the DB2 server system to create the necessary databases:

**Notes:**

- Replace `dbname` with the actual name of the database. Run the commands and each time replace `dbname` with the actual values for release, community, customization, Java Content Repository, Feedback, and Likeminds. You will need to run the commands once for each database for a total of six times.

**Remember:** DB2 database names cannot
exceed eight characters. Therefore, consider using these database names: release, commun, custom, jcrdb, fdbkdb, and lmdb.

db2 "CREATE DB dbname using codeset UTF-8 territory us PAGESIZE 8192"
db2 "UPDATE DB CFG FOR dbname USING applheapsz 4096"
db2 "UPDATE DB CFG FOR dbname USING app_ctl_heap_sz 1024"
db2 "UPDATE DB CFG FOR dbname USING stmtheap 32768"
db2 "UPDATE DB CFG FOR dbname USING dbheap 2400"
db2 "UPDATE DB CFG FOR dbname USING locklist 1000"
db2 "UPDATE DB CFG FOR dbname USING logfilsiz 4000"
db2 "UPDATE DB CFG FOR dbname USING logprimary 12"
db2 "UPDATE DB CFG FOR dbname USING logsecond 20"
db2 "UPDATE DB CFG FOR dbname USING logbufsz 32"
db2 "UPDATE DB CFG FOR dbname USING avg_appls 5"
db2 "UPDATE DB CFG FOR dbname using AUTO_MAINT off"

Complete the following:

6. On the DB2 server system, run the following commands. This step is only required for the IBM Java Content Repository database (jcrdb).

   - jcrdb is the name of the database used to store user data and objects
   - jcr is the jcr user for jcrdb
   - dbpassword is the password for the jcr user for the jcrdb

```
db2 "CONNECT TO jcrdb USER jcr USING dbpassword"
db2 "CREATE BUFFERPOOL ICMLSFREQBP4 SIZE 1000 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL ICMLSVOLATILEBP4 SIZE 16000 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL ICMLSMAINBP32 SIZE 16000 PAGESIZE 32 K"
db2 "CREATE BUFFERPOOL CMBMAIN4 SIZE 1000 PAGESIZE 4 K"
db2 "CREATE REGULAR TABLESPACE ICMLFQ32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLFQ32') BUFFERPOOL ICMLSMAINBP32"
db2 "CREATE REGULAR TABLESPACE ICMLNF32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLNF32') BUFFERPOOL ICMLSMAINBP32"
db2 "CREATE REGULAR TABLESPACE ICMVFQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('ICMVFQ04') BUFFERPOOL ICMLSVOLATILEBP4"
db2 "CREATE REGULAR TABLESPACE ICMSFQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('ICMSFQ04') BUFFERPOOL ICMLSFREQBP4"
db2 "CREATE REGULAR TABLESPACE CMBINV04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('CMBINV04') BUFFERPOOL CMBMAIN4"
db2 "CREATE SYSTEM TEMPORARY TABLESPACE ICMLSSYSTSPACE32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLSSYSTSPACE32') BUFFERPOOL ICMLSMAINBP32"
db2 "CREATE SYSTEM TEMPORARY TABLESPACE ICMLSSYSTSPACE4 PAGESIZE 4 K MANAGED BY SYSTEM USING ('ICMLSSYSTSPACE4') BUFFERPOOL ICMLSVOLATILEBP4"
```

db2 "DISCONNECT jcrdb"
db2 "TERMINATE"

B. On the DB2 server system, check the services file. If it does not specify DB2 connection and interrupt service ports, specify the ports for your operating system: Use a text editor to open the file /etc/services and add the following text: db2c_db2inst1port1/tcp

where db2inst1 is the default instance and port1 is the TCP port DB2 listens on. Replace port1 with the actual port.
number that assigned to the DB2 connection service in your DB2 server installation.
The `/etc/services` file is located under `%systemroot%/system32/drivers/`, where `%systemroot%` is the location of the operating system. For example: `C:\Windows\system32\drivers\etc\services`.

C. For JDBC Type 2 connections only: On DB2 Connect, use a text editor to open the `/etc/services` file. If it does not specify the DB2 connection service port, add the following text to specify the port for the remote DB2 instance:

```
db2c_db2inst1.port1/tcp # DB2 connection service port
```

where `db2inst1` is the name of the DB2 instance ID on the system, and `port1` with the actual port number that is assigned to the DB2 connection service in your DB2 server installation. The connection service port on the DB2 Client system, WebSphere Portal server, must match the connection service port on the DB2 server. The ports should match by number but not necessarily by name.

D. Set up the correct service name by entering the following command on the DB2 server system:

```
db2 "UPDATE DBM CFG USING svcename svce_name" where svce_name is the connection service port name that is specified in substep b, such as `db2c_db2inst1`.
```

E. For JDBC Type 2 connections only: On DB2 Connect, set DB2COMM to TCP/IP by using the `db2set` command:
```
db2set DB2COMM=tcpip.
```

F. For JDBC Type 2 connections only: Catalog the TCP/IP node with the IP address of the remote database server on DB2 Connect:
```
db2 "catalog tcpip node remote_db_node_alias remote database_server_node connection_service_port" where:

- `remote_db_node_alias` is the alias name of the database server that you are defining for the WebSphere Application Server node name. The alias name can contain one to eight characters.
- `database_server_node` is the fully qualified host name of your database server system.
- `connection_service_port` is the name of the DB2 connection service port that is configured in the `/etc/services` file on the database server system.
```

G. For JDBC Type 2 connections only, catalog the WebSphere Portal databases on DB2 Connect, where:

- `remote_db_name_domain`, is the cataloged name of the databases on the server system for each domain.
- `domain_alias_name`, is the database alias names that you are defining.
- `remote_db_node_alias` is the name that was used previously when you cataloged the TCP/IP node in the previous step.

The alias for each database must be different from the actual database name and can only contain up to eight characters:
```
db2 "catalog db remote_db_name_release as release_alias_name at node remote_db_node_alias"
db2 "catalog db remote_db_name_community as comm_alias_name at node remote_db_node_alias"
db2 "catalog db remote_db_name_customization as cust_alias_name at node remote_db_node_alias"
```

```
db2 "catalog db remote_db_name_fdbkdb as fdbkdb_alias_name at node remote_db_node_alias"
db2 "catalog db remote_db_name_lmdb as lmdb_alias_name at node remote_db_node_alias"
db2 "catalog db remote_db_name_jcrdb as jcrdb_alias_name at node remote_db_node_alias"
```

H. For JDBC Type 2 connections only: Log out of DB2 Connect by entering `db2 "terminate"`.

I. For JDBC Type 2 connections only, on DB2 Connect, test your remote connection by issuing the following command in the DB2 command window:
```
db2 "connect to alias_name user username using password", where alias_name is the alias name that you defined above, username is the database user, and password is the password assigned to the database user.
```

Parent topic: Setting up a remote DB2 database
Previous topic: Create users
Next topic: Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the \wp_profile_root\PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- The page size of table spaces used by WebSphere Portal must be 8192 bytes.
- For details on creating table spaces refer to the documentation for the database.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file \wp_profile_root\PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.index_name.indexspace
   For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
     - release
     - community
     - customization
     - jcr
     - feedback
     - likeminds
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword IN and a space. For example: community.COMP_INST.tablespace=IN COMM8KSPACE
   Repeat this step for each domain that you are transferring.
4. Save and close dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
   - UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

Parent topic: Setting up a remote DB2 database
Previous topic: Creating remote databases
Next topic: Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

- Prerequisites
- Installing DB2
- Create users
- Creating remote databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

Tip: For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Working with properties files:
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:

  - release.DbName=release
  - jcr.DbName=jcrdb
  - feedback.DbName=fdbkdb
  - likeminds.DbName=lmdb
  - community.DbName=commdb
  - customization.DbName=custdb

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:

  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
- **dbdomainDbType**
- **dbdomainDbName**
- **dbdomainDbUrl**
- **dbdomainDbSchema**

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If **DbUser**, **DbUrl**, and **DbPassword** are not the same across domains, the value for **DataSourceName** must differ from the **DataSourceName** of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   - `wp_profile_root/ConfigEngine/properties/wkplc.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_comp.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties`

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In `wkplc_comp.properties`, most properties are repeated for each domain.

2. Use a text editor to open the properties file `wkplc_comp.properties` and modify the values to correspond to your environment. **Note:** The `wkplc_comp.properties` file by default has the properties populated for the Apache Derby database. The source `dbdomain.properties` represent the properties for the source database. These values should not be modified unless you are transferring from a supported database other than Apache Derby.

   A. For **dbdomainDbType**, type `db2`.
   B. For **dbdomainDbName**, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the `dbdomain.DbUrl` property.
   C. For **dbdomainDbName**, type the name of the WebSphere Portal domain database. DB2 database names cannot exceed eight (8) characters. **Note:** This value is also the database element in the `dbdomain.DbUrl` property.
   D. For **dbdomainDbSchema**, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   E. For **dbdomainDataSourceName**, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - `releaseDS`
      - `communityDS`
      - `customizationDS`
      - `jcrDS`
      - `lmdbDS`
      - `feedback`
   F. For **dbdomainDbUrl**, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:** The database element of this value should match the value of `DbName`.
   G. For **dbdomainDbUser**, type the user ID for the database administrator.
   H. For **dbdomainDbPassword**, type the password for the database administrator.
   I. Optional: For **dbdomainDbRuntimeUser**, type the user ID of the database user that should be used by WebSphere Portal to connect to the database at runtime.
   J. If **dbdomainDbRuntimeUser** is specified, you must set **dbdomainDbRuntimePassword** to be the password of the runtime database user.
   K. For **dbdomainDBA.DbUser**, type the database administrator user ID for privileged access operations during creation of the database.
L. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

   A. For `source.domainDbType`, type of the database you are currently configured to use. The value for `source.domainDbType` is Derby by default.

   B. For `source.domainDbName`, type the name of the database domain you are currently using.

   C. For `source.domainDbSchema`, type current schema identifier for objects within the database for this domain.

   D. For `source.domainDataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

   E. For `source.domainDbUrl`, type the url currently used to access your database.

   F. For `source.domainDbUser`, type the name of the user accessing this database.

   G. For `source.domainDbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

   A. For `db2.DbDriver`, type the name of the JDBC driver class.

   B. For `db2.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For `db2.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Setting up a remote DB2 database  
**Previous topic:** Assigning custom table spaces  
**Next topic:** Configuring JCR collation support
Configuring JCR collation support

View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties

1. Stop the WebSphere Portal server:
   ```
   stopServer.bat WebSphere_Portal -username admin_userid -password admin_password
   ```
2. If using a remote DB2 server, copy the following files from the WebSphere Portal server to a temporary directory on the remote DB2 server:
   ```
   PortalServer_root/jcr/prereq.jcr/config/collation.jar
   PortalServer_root/jcr/prereq.jcr/config/registerCollationUDFTemplate.sql
   ```
3. Set up collation on the database where the JCR domain is located. Log in to the database machine with a userid that is authorized and configured to use the appropriate DB2 instance. For example, a common userid is `db2inst1`.
   A. Change to the directory `db2home/function`.
   B. Execute the command:
      ```
      Remote DB2: db2home/java/jdk/bin/jar -xvf temporary location/collation.jar
      icm/CollationUDF.class
      ```
      Local DB2: `db2home\java\jdk\bin\jar -xvf PortalServer_root\jcr\prereq.jcr\config\collation.jar
      icm/CollationUDF.class`
   C. Change to the directory `wp_profile_root\PortalServer\jcr\config`. If using a remote DB2, change to the temporary directory where you copied the files in a previous step.
   D. Open the file `registerCollationUDFTemplate.sql` and change all `SCHEMA` references to the JCR schema; for example, `JCR`. The value set for `SCHEMA` should match the value set for the `jcr.DbSchema` property.
   E. Connect to the JCR database by running `db2 connect to <jcrdb> user <userid> using <password>`. You must connect to the JCR database using the userid and password of the WebSphere Portal JCR data source.
   F. Execute the following command:
      ```
      Remote DB2: db2 -tvf temporary location/registryCollationUDFTemplate.sql
      Local DB2: db2 -tvf wp_profile_root\PortalServer\jcr\config\registerCollationUDFTemplate.sql
      ```
   G. Disconnect from the JCR database.
   H. Restart the DB2 instance.
4. Verify that the UDF is registered properly.
   A. Log in as the `db2instanceID`. Open a DB2 terminal window, and type `db2`. From the command line, type the command `connect to <jcrdb> user <userid> using <password>`. You must connect to the JCR database using the userid and password of the WebSphere Portal JCR data source.
   B. Execute the following command in the previous substep. If the command completes successfully, the UDF is registered correctly as follows: values `schema.sortkeyj('abc','en')`, where `schema` is the schema used in the previous substep.
   C. Edit the `icm.properties` file, located in `wp_profile_root\PortalServer\jcr\lib\com\ibm\icm` directory. Add the following section to the end of the file:
      ```
      # Enable/Disable collation support for all DB2 platforms
      ```
# Disabled by default
jcr.query.collation.db2.enabled = true

# Database specific collation mappings
# These mappings apply map a Java locale name into a collation name
# supported by the underlying database.
# Example mappings for DB2 platform

# English
jcr.query.collation.en = en

# Swedish
jcr.query.collation.sv = sv

jcr.query.collation.zh = zh
jcr.query.collation.de = de
jcr.query.collation.da = da
jcr.query.collation.hu = hu
jcr.query.collation.jp = jp

6. Enter the following command to start the WebSphere Portal server: startServer.bat WebSphere_Portal

Parent topic: Setting up a remote DB2 database
Previous topic: Modifying database properties
Next topic: Configuring WebSphere Portal to use DB2
Configuring WebSphere Portal to use DB2

View information on manually transferring data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties

Tips:
- To run these tasks as a non-root user, you must first run the task chown -R non-root_userWebSphereDir.
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.
- Be sure that DB2 is started by checking the service. If attempts to restart result in a logon failure message, then go to the DB2 properties and reenter the password.

Steps for transferring data to another supported database

1. If you are running a type 2 connection, before transferring data edit the db2cli.ini file. Failure to follow these steps will cause the database transfer to hang at the task action-process-constraints.
   A. Locate the file C:\Program Files\IBM\SQLLIB\db2cli.ini.
   B. Add the following to the end of the file. Leave an empty line after ReturnAliases=0.[COMMON]

   DYNAMIC=1
   ReturnAliases=0

2. Open a command prompt and change to the directory wp_profile_root\ConfigEngine.
3. Enter the following commands to validate configuration properties.
   ConfigEngine.bat validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ConfigEngine.bat validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
4. From the same command prompt as the previous steps, change to the directory wp_profile_root\bin.
5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td><code>stopServer.bat server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td><code>stopServer.bat WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

6. Transfer the database:

**Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

A. Change to the directory `wp_profile_root\ConfigEngine`.

B. Enter the following command:

```
ConfigEngine.bat database-transfer -DWasPassword=password
```

**Note:**
- To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:

```
ConfigEngine.bat database-transfer -DTransferDomainList=jcr -DWasPassword=password
```
- If you have been storing data in IBM® Derby for a long time, database transfer could fail with OutOfMemory exceptions. If database transfer fails, add the following property to the command in this step:

```
ConfigEngine.bat database-transfer -DDbtJavaMaxMemory=1536M -DWasPassword=password
```

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

7. After transferring the database tables, perform a `reorg` check to improve performance. Perform this step for each database alias in the property file.

A. Connect to a database with the following command:

```
db2 connect to database_alias user db2admin_userid using password
```

**Note:** Additional options might be required if additional security has been installed. Refer to DB2 Universal Database commands by example for links to the command reference.

B. After it is connected, run the following command from the DB2 prompt:

```
db2 reorgchk update statistics on table all > xyz.out
```

C. Look in the `reorg` column for entries marked with a `*` (star or asterisk) in the file `xyz.out`. For each line with a `*`, note the `tablename` and run the following command for each `tablename`:

```
db2 terminate
db2rbind database_name -l db2rbind.out -u db2_admin -p password
```

D. The output file `db2rbind.out` is only created when there is an error for the `db2rbind` command.

8. Change to the directory `wp_profile_root\bin`.

9. Enter the following command to start the WebSphere Portal server:

```
startServer.bat WebSphere_Portal
```

Parent topic: Setting up a remote DB2 database

Previous topic: Configuring JCR collation support

Next topic: Configuring DB2 for large file handling in Web Content Management
Configuring DB2 for large file handling in Web Content Management

If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties
  - Configuring WebSphere Portal to use DB2

**Note:** You only need to perform these steps if you are using Web Content Management.

1. Log into the WebSphere Application Server administrative console.
2. Click **Resources > JDBC > Data sources**.
3. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
4. Select the name of the data source that is defined in `wkplc_comp.properties` for the JCR database domain. The default data source is `wpdbDS`.
5. Click **Custom properties**.
6. Ensure that the `fullyMaterializeLobData` property is set to `false`.

**Parent topic:** Setting up a remote DB2 database  
**Previous topic:** Configuring WebSphere Portal to use DB2  
**Next topic:** Changing driver types
Changing driver types

WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Before you begin, ensure that the following conditions are met:
- The WebSphere Portal database has been successfully transferred to DB2 using the `database-transfer` configuration task.
- The files `wkplc_comp.properties` and `wkplc_dbtype.properties` have been modified to set the correct values for the DB2 drivers that you are switching to: In the file `wkplc_comp.properties` set each `<Domains>.DbUrl` property using the following formats:

  # db2 (type 2):        { jdbc:db2:wpsdb }
  # db2 (type 4):        { jdbc:db2://<YourDatabaseServer>:50000/wpsdb:returnAlias=0; }

In the file `wkplc_dbtype.properties` set the `db2.DbLibrary` property using the following format: For DB2 Type 2 driver use `<SQLLIB>/java/db2java.zip`

  # For DB2 Type 2 driver use <SQLLIB>/java/db2java.zip
  # For DB2 Type 4 driver use <SQLLIB>/java/db2jcc.jar;<SQLLIB>/java/db2jcc_license_cu.jar

In the file `wkplc_dbtype.properties` set the `db2.DbDriver` property using the following format: For DB2 Type 2 driver use `COM.ibm.db2.jdbc.app.DB2Driver`

  # For DB2 Type 2 driver use COM.ibm.db2.jdbc.app.DB2Driver
  # For DB2 Type 4 driver use com.ibm.db2.jcc.DB2Driver

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties
  - Configuring WebSphere Portal to use DB2
  - Configuring DB2 for large file handling in Web Content Management

1. Open a command prompt and change to the directory `wp_profile_root\ConfigEngine`. Command: `stopServer.bat server1 -username admin_userid -password admin_password`

2. Enter the following commands to validate configuration properties. Command: `ConfigEngine.bat validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password`

3. From the same command prompt as the previous steps, change to the directory `wp_profile_root\bin`. Command: `stopServer.bat WebSphere_Portal -username admin_userid -password admin_password`

4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td><code>stopServer.bat server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td><code>stopServer.bat WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>


6. To change from one supported driver to the other, run the following task to connect the database, including only the domains that require the switch. Command: `ConfigEngine.bat connect-database -Drelease.DbPassword=password`
-Dlikeminds.DbPassword=password  -DWasPassword=password

7. Change to the directory `wp_profile_root\bin`.
8. Enter the following command to start the WebSphere Portal server:
```bash
startServer.bat WebSphere_Portal
```

**Parent topic:** Setting up a remote DB2 database

**Previous topic:** Configuring DB2 for large file handling in Web Content Management
Setting up a remote Oracle database on Windows for a stand-alone production server

Setting up the Oracle database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

1. **Installing Oracle**
   - View information on how to install Oracle for use with WebSphere Portal.

2. **Creating users**
   - View the steps to set up users for Oracle to work with WebSphere Portal.

3. **Creating databases**
   - View some important considerations before setting up Oracle databases to work with WebSphere Portal.

4. **Modifying database properties**
   - Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. **Setting up databases**
   - This topic provides instructions on automatically setting up your database using the ConfigEngine task to create users, grant permissions, and create Java Content Repository table spaces.

6. **Windows stand-alone server: Creating JCR table spaces**
   - This topic provides manual instructions for creating JCR table spaces for Oracle.

7. **Optional: Assigning custom table spaces**
   - The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

8. **Configuring WebSphere Portal to use Oracle**
   - View information on manually transferring data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

**Parent topic:** Setting up a database
Installing Oracle

View information on how to install Oracle for use with WebSphere Portal.
You can use Oracle as the database software. You must install Oracle separately and before performing a database transfer.
Refer to Oracle product documentation for installation instructions.

Parent topic: Setting up a remote Oracle database on Windows for a stand-alone production server
Next topic: Creating users
Creating users

View the steps to set up users for Oracle to work with WebSphere Portal.

Before you begin:

- You should have installed Oracle.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

**Prerequisites**

- Installing Oracle

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Start the Oracle SQL command line processor 'SQL*Plus' by entering `sqlplus /nolog` on the operating system command prompt

2. Log in to the Oracle database by executing command `connect user/password@database name`. For example: `system/manager@wpsdb` will log the administrative user system with a password of manager into the wpsdb database.

3. Create the WebSphere Portal user `dbdomain.DbUser`, where `dbdomain` is replaced by release, community, or customization.

   A. Log in to the database in which you want to create the new users.
   
   B. Create the users that you need in the appropriate databases by entering the following commands in the SQL*Plus tool, where `user_tablespace` is the default tablespace that is identified by the database administrator to store user objects and `temp_tablespace` is identified to store temporary objects. All names must be in lower case.

   ```
   SQL> create user releaseusr identified by password
       default tablespace user_tablespace
       temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO releaseusr;
   SQL> create user communityusr identified by password
       default tablespace user_tablespace temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO communityusr;
   SQL> create user customizationusr identified by password
       default tablespace user_tablespace temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO customizationusr;
   SQL> create user jcr identified by password
       default tablespace user_tablespace temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO jcr
   ```

4. Connect to the Feedback database:

   A. Enter the following command: `SQL> connect`
B. Enter *user-name: username/password@dbname* where *username* is an existing administrative user in the database. For example, *system/manager@fdbkdb* will log the administrative user system with a password of manager into the *fdbkdb* database.

C. Create the Feedback user:

```sql
SQL> create user feedback identified by password
    default tablespace user_tablespace
    temporary tablespace temp_tablespace;
```

GRANT UNLIMITED TABLESPACE TO *feedback*

D. Log out of the command line tool using the command *SQL> exit*.

5. Connect to the LikeMinds database:

A. Enter the following command:

```sql
SQL> connect
```

B. Enter *user-name: username/password@dbname*, where *username* is an existing administrative user in the database.

For example, *system/manager@lmdb* will log the administrative user system with a password of manager into the *lmdb* database.

C. Create the LikeMinds user:

```sql
SQL> create user lmdbusr identified by password
    default tablespace user_tablespace
    temporary tablespace temp_tablespace;
```

GRANT UNLIMITED TABLESPACE TO *lmdbusr*

D. Log out of the command line tool using the command *SQL> exit*.

6. Perform the following steps to verify that you can connect to the database with the user ID and password that you just created:

A. Enter the *SQL> connect* command to connect to the content database.

B. Enter *user-name: username/password@dbname*, where *username* is an existing administrative user in the database. For example, *system/manager@jcrdb* will log the administrative user system with a password of manager into the Oracle database.

**Parent topic:** Setting up a remote Oracle database on Windows for a stand-alone production server  
**Previous topic:** Installing Oracle  
**Next topic:** Creating databases
Creating databases

View some important considerations before setting up Oracle databases to work with WebSphere Portal.

For information about creating databases, refer to the Oracle product documentation. For information on the recommended database architecture and the databases you will need to create, see the Planning for Oracle topic. Be sure that all databases to be used with WebSphere Portal are created as UNICODE character set databases.

If you are using remote Oracle 9 or 10 databases, you must also copy the ojdbc14.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM® WebSphere® Application Server Version 7.0, you must also copy the ojdbc6.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM WebSphere Application Server Version 6.1, you must also copy the ojdbc5.jar file from the remote Oracle server to the WebSphere Portal machine. The typical location is the oracle_home/sqldeveloper/jdbc/lib directory. Record the copy location on your local machine for future reference.

When creating Oracle databases for use with WebSphere Portal, you should consider the following information:

- The Oracle databases must be created manually before configuring WebSphere Portal.
- All databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle server will be remote or local to the WebSphere Portal installation.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle database in order for WebSphere Portal to communicate with the Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes:
  - $\text{db\_block\_size} = 8192$ bytes
  - $\text{db\_cache\_size} = 307,200$ bytes
  - $\text{db\_files} = 1024$ files
  - $\text{log\_buffer} = 65536$ bytes
  - $\text{open\_cursors} = 1500$ cursors
  - $\text{pga\_aggregate\_target} = 204,800$ bytes
  - $\text{pre\_page\_sga} = \text{true}$
  - $\text{processes} = 300$ processes
  - $\text{shared\_pool\_size} = 204,800$ bytes

  **Note:** If you are using IBM Java Content Repository, the $\text{open\_cursors}$ value may need to be increased based on the table count in the Java Content Repository schema.

- Raise the number of parallel servers as appropriate. For example, if you have more than 875 parallel servers, you should set the $\text{parallel\_max\_servers}$ to 1200.

- The Oracle parameter CURSOR_SHARING allows similar SQL Statements to be shared when possible, which prevents parsing and establishing a new execution plan. The execution plan is used by Oracle to gather the data needed to satisfy a request. There are two options for CURSOR_SHARING, which are as follows:
  - **FORCE**
    - When you select this option, Oracle uses the same execution plan for all SQLs that are similar in value even if the values are different. When you use this option, the execution plan may not provide optimum performance. For example, similar SQLs with different values may behave differently when executed running the same plan.
  - **EXACT**
- When you select this option, Oracle only shares the same execution plan for SQLs that are identical and use the same values. This option removes the risk of a SQL statement being executed when optimum performance conditions do not exist.

WebSphere Portal supports both options. Regardless of the option selected, portlet applications should not be affected. Contact your database administrator for further assistance on these options.

- **Prerequisites**
  - Installing Oracle
  - Creating users

**Parent topic:** Setting up a remote Oracle database on Windows for a stand-alone production server  
**Previous topic:** Creating users  
**Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Installing Oracle
  - Creating users
  - Creating databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:
  - `release.DbName=release`
  - `jcr.DbName=jcrdb`
  - `feedback.DbName=fdbkdb`
  - `likeminds.DbName=lmdb`
  - `community.DbName=commdb`
  - `customization.DbName=custdb`

  - If you are using a remote database, enter the values for the remote server.
  - Use a forward slash (/) instead of a backslash (\).
  - There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
  - The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
  - Depending on which database domain has to be configured, replace `dbdomain` with:
    - `release`
    - `customization`
    - `community`
    - `jcr`
    - `feedback`
    - `likeminds`

  - The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
    - `dbdomain.DbType`
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.

   A. For dbdomainDbType, type oracle.
   B. For dbdomainDbName, type the name of the WebSphere Portal domain database. Note: This value is also the database element in the dbdomain_DbUrl property.
   C. For dbdomainDbSchema, type the schema name of the database domain. Note: Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The dbdomainDbName should be the same value used for the dbdomain.dbSchema

      Restriction: The value for dbdomainDbSchema must equal the value for dbdomainDbUser.
   D. For dbdomainDataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:

      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback

   E. For dbdomainDbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. Note:

      - The database element of this value should match the value of DbName.
      - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following:

         jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.

   F. For dbdomainDbUser, type the user ID for the database administrator. Restriction: The value for dbdomainDbUser must equal the value for dbdomainDbSchema.
G. For dbdomain..DbPassword, type the password for the database administrator.

H. For dbdomain=DBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.

I. For dbdomain=DBA.DbPassword, type the database administrator password for privileged access operations during creation of the database.

J. For dbdomain.DbHome, type the root location for the database. **Note:** This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For source.domainDbType, type of the database you are currently configured to use. The value for source.domainDbType is Derby by default.

B. For source.domainDbName, type the name of the database domain you are currently using.

C. For source.domainDbSchema, type current schema identifier for objects within the database for this domain.

D. For source.domain.DataSourceName, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For source.domainDbUrl, type the url currently used to access your database.

F. For source.domainDbUser, type the name of the user accessing this database.

G. For source.domainDbPassword, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file wkplcDbType.properties.

A. For oracle.DbDriver, type the name of the class that SqlProcessor uses to import SQL files.

B. For oracle.DbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

C. For oracle.JdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file wkplc.properties.

A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Setting up a remote Oracle database on Windows for a stand-alone production server  
**Previous topic:** Creating databases  
**Next topic:** Setting up databases
Setting up databases
This topic provides instructions on automatically setting up your database using the ConfigEngine task to create users, grant permissions, and create Java Content Repository table spaces.

**Prerequisites**
- Installing Oracle
- Creating users
- Creating databases
- Modifying database properties

As an alternative to automatically setting up the database, you can manually set up your database by referring to the link in the related tasks section of this topic.

1. On the database server, make sure that the folder `db_1` contains the subfolders `your_oracle_instance/data` and `your_oracle_instance/index`. If this folder hierarchy does not exist, create it manually before you run the `setup-database` task. The `setup-database` task requires these folders to create database users. If these folders do not exist, the `setup-database` task will fail.
2. Change to the directory `wp_profile_root/ConfigEngine`
3. To create the database users, type the following command:

   ```
   ConfigEngine.bat setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:

   ```
   <WP_root>/base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql
   ```

**Parent topic:** Setting up a remote Oracle database on Windows for a stand-alone production server

**Previous topic:** Modifying database properties

**Next topic:** Windows stand-alone server: Creating JCR table spaces

**Related tasks**
Manually creating users and granting privileges for Oracle
Windows stand-alone server: Creating JCR table spaces

This topic provides manual instructions for creating JCR table spaces for Oracle.

**Prerequisites**
- Installing Oracle
- Creating users
- Creating databases
- Modifying database properties
- Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.
2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:
   - `&jcrdb` is the name of the database you created to store user data.
   - `&dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.

   Ensure that the `.` is included in the variables when you substitute the values of your environment with these variables.

**Important:** You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

```
cREATE TABLESPACE ICMLFQ32 DATAFILE '&dbpath./&jcrdb./data/ICMLFQ32_01.dbf' SIZE 300M REUSE AUTOEXTEND NEXT 10M MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
CREATE TABLESPACE ICMLNF32 DATAFILE '&dbpath./&jcrdb./data/ICMLNF32_01.dbf' SIZE 25M REUSE AUTOEXTEND NEXT 10M MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
CREATE TABLESPACE ICMVFQ04 DATAFILE '&dbpath./&jcrdb./data/ICMVFQ04_01.dbf' SIZE 25M REUSE AUTOEXTEND NEXT 10M MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
CREATE TABLESPACE ICMSFQ04 DATAFILE '&dbpath./&jcrdb./data/ICMSFQ04_01.dbf' SIZE 150M REUSE AUTOEXTEND NEXT 10M MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
CREATE TABLESPACE ICMLSNDX DATAFILE '&dbpath./&jcrdb./index/ICMLSNDX_01.dbf' SIZE 10M REUSE AUTOEXTEND NEXT 10M MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
```

A. Set the size, autoextend, and maxsize values according to your environment. For example, you may want to change the maxsize to a set value rather than UNLIMITED.

B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.

C. Refer to the Oracle command reference for more information about using the `create tablespaces` command.

**Parent topic:** Setting up a remote Oracle database on Windows for a stand-alone production server

**Previous topic:** Setting up databases

**Next topic:** Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the \wp_profile_root\PortalServer\config\tablespaces\dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

- Prerequisites
  - Installing Oracle
  - Creating users
  - Creating databases
  - Modifying database properties
  - Setting up databases
  - Windows stand-alone server: Creating JCR table spaces

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file \wp_profile_root\PortalServer\config\tablespaces\dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - \dbdomain.table_name.tablespace
   - \dbdomain.table_name.index_name.indexspace
   For the file name and each table space and index space property pair, \dbdomain\ can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword \TABLESPACE and a space. For example: \community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE
   Repeat this step for each domain that you are transferring.
4. Save and close \dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
Configuring WebSphere Portal to use Oracle

View information on manually transferring data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- **Prerequisites**
  - Installing Oracle
  - Creating users
  - Creating databases
  - Modifying database properties
  - Setting up databases
  - Windows stand-alone server: Creating JCR table spaces

**Tips:**

- If you are transferring from Oracle, the `open_cursors` setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

**Steps for transferring data to another supported database**

1. Open a command prompt and change to the directory `wp_profile_root\ConfigEngine`.
2. Enter the following commands to validate configuration properties.
   ```plaintext
   ConfigEngine.bat validate-database-driver
   ConfigEngine.bat validate-database-connection
   ```
3. From the same command prompt as the previous steps, change to the directory `wp_profile_root\bin`.
4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td>stopServer.bat server1 -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>
5. Transfer the database:

   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory `wp_profile_root\ConfigEngine`.

   B. Enter the following command:

       ```
       ConfigEngine.bat database-transfer -DWasPassword=password
       ```

       **Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:

       ```
       ConfigEngine.bat database-transfer -DTransferDomainList=jcr -DWasPassword=password
       ```

   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

       If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example:

   ```
   SQL> execute dbms_stats.gather_schema_stats(ownname=> 'jcr', cascade=> TRUE);
   ```

7. Change to the directory `wp_profile_root\bin`.

8. Enter the following command to start the WebSphere Portal server:

   ```
   startServer.bat WebSphere_Portal -username admin_userid -password admin_password
   ```

---

**Parent topic:** Setting up a remote Oracle database on Windows for a stand-alone production server

**Previous topic:** Assigning custom table spaces
Setting up a remote Oracle RAC database on Windows for a stand alone production server

Setting up the Oracle RAC database includes creating user IDs and databases on a remote server. A task is provided to assist with creating the users. Before you can use the task, you must modify properties files.

1. **Installing Oracle RAC**
   View information on how to install Oracle RAC for use with WebSphere Portal.

2. **Create users**
   View the steps to set up users for Oracle RAC to work with WebSphere Portal.

3. **Creating databases**
   View some important considerations before setting up Oracle RAC databases to work with WebSphere Portal.

4. **Modifying database properties**
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. **Setting up databases**
   This topic provides instructions on automatically setting up your database using the ConfigEngine task to create users, grant permissions, and create Java Content Repository table spaces. As an alternative to automatically setting up the database, you can manually set up your database.

6. **Windows stand-alone server: Creating JCR table spaces**
   This topic provides manual instructions for creating JCR table spaces for Oracle.

7. **Optional: Assigning custom table spaces**
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

8. **Configure WebSphere Portal to use Oracle RAC**
   View information on manually transferring data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

**Parent topic:** Setting up a database
Installing Oracle RAC

View information on how to install Oracle RAC for use with WebSphere Portal.

Before you begin:
- You should have completed reviewing the Planning for Oracle RAC section.
- Ensure the Oracle CRS (Cluster Ready Service) and Oracle RAC databases have been installed and configured on the primary and secondary nodes.
- Run the following commands on both RAC nodes to start global services daemon (GSD), oracle listeners, and agents.
  - `$ gsdctl start`
  - `$ lsnrctl start`
  - `$ agentctl start`

You can use Oracle RAC as the database software for WebSphere Portal. You must install Oracle RAC separately and before performing WebSphere Portal database transfer.

Refer to Oracle product documentation for installation instructions.

Parent topic: Setting up a remote Oracle RAC database on Windows for a stand alone production server

Next topic: Create users

Related tasks
Manually creating users and granting privileges for Oracle
Create users

View the steps to set up users for Oracle RAC to work with WebSphere Portal.

Before you begin:
- You should have completed installing Oracle RAC.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

**Prerequisites**
- Installing Oracle RAC

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Create the users in the WebSphere Portal databases
   A. Log in to the database in which you want to create the new users.
   B. Create the users that you need in the appropriate databases by entering the following command in the SQL*Plus tool, where `user_tablespace` is the default tablespace that is identified by the database administrator to store user objects and `temp_tablespace` is identified to store temporary objects:

   ```sql
   SQL> create user username identified by password
   default tablespace user_tablespace
   temporary tablespace temp_tablespace;
   ```

   Tip: Balance the storage of user objects among tablespaces to prevent running out of space with overuse of `user_tablespace`. Also consider increasing the size of `user_tablespace` when handling a high volume of users.

2. Log in by entering the command `sqlplus` in SQL*Plus:

3. Enter the command `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example, `system/manager@wpsdb` will log the administrative user system with a password of manager into the wpsdb database.

4. Create the WebSphere Portal user `dbdomain.DBUser`, where `dbdomain` is replaced by release, community, customization, or JCR. All names must be in lower case. The following code uses the default user tablespace and the default temporary tablespace. Explicitly specify these tablespaces if you do not want to use the defaults:

   ```sql
   SQL> create user releaseusr identified by password;
   SQL> create user communityusr identified by password;
   SQL> create user customizationusr identified by password;
   SQL> create user jcr identified by password;
   ```

5. Connect to the content database by entering the command `SQL> connect`.

6. Enter the following, where `username` is an existing administrative user in the database.

   ```sql
   user-name: username/password@dbname
   ```

   For example, `system/manager@jcrdb` will log the administrative user system with a password of manager into the Oracle RAC database.

7. Create the Oracle RAC users by entering the following: When creating the `jcr` user, grant all necessary privileges. If you do not grant privileges, you will receive the error `ICMADMIN lacks CREATE SESSION Privilege logon denied`. 

   ```sql
   SQL> connect username/password@dbname
   ```
when you try to connect with the \texttt{jcruser}: \texttt{SQL} > create user \texttt{jcr} identified by \texttt{password} default tablespace users temporary tablespace temp;

8. Connect to the Feedback database:
   A. Enter the following command: \texttt{SQL} > connect
   B. Enter \texttt{user-name}: \texttt{username/password@dbname} where \texttt{username} is an existing administrative user in the database.
      For example, system/manager@fdbkdb will log the administrative user system with a password of manager into the fdbkdb database.

9. Create the Feedback user: \texttt{SQL} > create user \texttt{feedback} identified by \texttt{password} default tablespace users temporary tablespace temp;

10. Connect to the LikeMinds database: \texttt{SQL} > connect
11. Enter \texttt{user-name}: \texttt{username/password@dbname}, where \texttt{username} is an existing administrative user in the database. For example, system/manager@lmdb will log the administrative user system with a password of manager into the lmdb database.
12. Create the LikeMinds user: \texttt{SQL} > create user \texttt{lmdbusr} identified by \texttt{password} default tablespace users temporary tablespace temp;
13. Log out of the command line tool using the command \texttt{SQL} > exit.

\textbf{Parent topic:} Setting up a remote Oracle RAC database on Windows for a stand alone production server  
\textbf{Previous topic:} Installing Oracle RAC  
\textbf{Next topic:} Creating databases
Creating databases

View some important considerations before setting up Oracle RAC databases to work with WebSphere Portal.

When creating Oracle RAC databases for use with WebSphere Portal, you should consider the following information:

- The Oracle RAC databases must be created manually before configuring WebSphere Portal.
- All databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle RAC server will be remote or local to the WebSphere Portal installation.
- If using an earlier version of Oracle RAC (9i or earlier), ensure that Oracle RAC JVM is also installed.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle RAC database in order for WebSphere Portal to communicate with the IBM Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes are:

  - `db_block_size = 8192` bytes
  - `db_cache_size = 314,572,800` bytes
  - `db_files = 1024` files
  - `log_buffer = 65536` bytes
  - `open_cursors = 1500` cursors
  - `pga_aggregate_target = 209,715,200` bytes
  - `pre_page_sga = true`
  - `processes = 300` processes
  - `shared_pool_size = 209,715,200` bytes

  **Note:** If you are using IBM Java Content Repository, the `open_cursors` value may need to be increased based on the table count in the IBM Java Content Repository schema.

**Prerequisites**
- Installing Oracle RAC
- Create users

- **Parent topic:** Setting up a remote Oracle RAC database on Windows for a stand alone production server
- **Previous topic:** Create users
- **Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Installing Oracle RAC
  - Create users
  - Creating databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release_DbName`, `jcr_DbName`, `feedback_DbName`, and `likeminds_DbName`. For example:
  - `release_DbName=release`
  - `jcr_DbName=jcrdb`
  - `feedback_DbName=fdbkdb`
  - `likeminds_DbName=lmdb`
  - `community_DbName=commdb`
  - `customization_DbName=custdb`
- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace `dbdomain` with:
  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
  - `dbdomainDbType`
- **dbdomain.DbName**
- **dbdomain.DbUrl**
- **dbdomain.DbSchema**

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If **DbUser**, **DbUrl**, and **DbPassword** are not the same across domains, the value for **DataSourceName** must differ from the **DataSourceName** of the other domains. In other words, this value must be unique for the database domain.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

1. Locate the following files and create a backup copy of each before changing any values:
   - `wp_profile_root/ConfigEngine/properties/wkplc.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_comp.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties`

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In `wkplc_comp.properties`, most properties are repeated for each domain.

2. Use a text editor to open the properties file `wkplc_comp.properties` and modify the values to correspond to your environment.

   A. For **dbdomainDbType**, type **oracle**.

   B. For **dbdomainDbName**, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the **dbdomainDbUrl** property.

   C. For **dbdomainDbSchema**, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The **dbdomain_DbName** should be the same value used for the **dbdomain_DbSchema**

      **Restriction:** The value for **dbdomain_DbSchema** must equal the value for **dbdomain_DbUser**.

   D. For **dbdomainDataSourceName**, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback

   E. For **dbdomainDbUrl**, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:**
      - The database element of this value should match the value of **DbName**.
      - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following:
        ```
        jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME.
        ```
      When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.

   F. For **dbdomainDbUser**, type the user ID for the database administrator. **Restriction:** The value for **dbdomainDbUser** must equal the value for **dbdomainDbSchema**.
G. For `dbdomain.DbPassword`, type the password for the database administrator.
H. For `dbdomain.DBA.DbUser`, type the database administrator user ID for privileged access operations during creation of the database.
I. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.
J. For `dbdomain.DbHome`, type the root location for the database. **Note:** This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.
   A. For `source.domain.DbType`, type the database you are currently configured to use. The value for `source.domain.DbType` is Derby by default.
   B. For `source.domain.DbName`, type the name of the database domain you are currently using.
   C. For `source.domain.DbSchema`, type current schema identifier for objects within the database for this domain.
   D. For `source.domain.DataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.
   E. For `source.domain.DbUrl`, type the url currently used to access your database.
   F. For `source.domain.DbUser`, type the name of the user accessing this database.
   G. For `source.domain.DbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.
   A. For `oracle.DbDriver`, type the name of the class that SqlProcessor uses to import SQL files.
   B. For `oracle.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.
   C. For `oracle.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.
   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Setting up a remote Oracle RAC database on Windows for a stand alone production server  
**Previous topic:** Creating databases  
**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task to create users, grant permissions, and create Java Content Repository table spaces. As an alternative to automatically setting up the database, you can manually set up your database.

Before you begin, ensure that the following prerequisites are met:

- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

**Prequisites**

- Installing Oracle RAC
- Create users
- Creating databases
- Modifying database properties

1. On the database server, make sure that the folder `db_1` contains the subfolders `your_oracle_instance/data` and `your_oracle_instance/index`. If this folder hierarchy does not exist, create it manually before you run the `setup-database` task. The `setup-database` task requires these folders to create database users. If these folders do not exist, the `setup-database` task will fail.

2. Change to the directory `wp_profile_root\ConfigEngine`

3. To create the database users, type the following command:

   ```bash
   ConfigEngine.bat setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:

   `<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql`

---

**Parent topic:** Setting up a remote Oracle RAC database on Windows for a stand alone production server

**Previous topic:** Modifying database properties

**Next topic:** Windows stand-alone server: Creating JCR table spaces

**Related tasks**

Manually creating users and granting privileges for Oracle
Windows stand-alone server: Creating JCR table spaces

This topic provides manual instructions for creating JCR table spaces for Oracle.

- **Prerequisites**
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties
  - Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.
2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:
   - `&jcrdb` is the name of the database you created to store user data.
   - `&dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.

   Ensure that the `.` is included in the variables when you substitute the values of your environment with these variables.

   **Important:** You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

   ```
   create tablespace ICMLFQ32 datafile '/&dbpath./&jcrdb./data/&jcrdb._ICMLFQ32_01.dbf' size 300M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLNF32 datafile '/&dbpath./&jcrdb./data/&jcrdb._ICMLNF32_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMVFQ04 datafile '/&dbpath./&jcrdb./data/&jcrdb._ICMVFQ04_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMSFQ04 datafile '/&dbpath./&jcrdb./data/&jcrdb._ICMSFQ04_01.dbf' size 150M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLSNDX datafile '/&dbpath./&jcrdb./index/&jcrdb._ICMLSNDX_01.dbf' size 10M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   
   A. Set the `size`, `autoextend`, and `maxsize` values according to your environment. For example, you may want to change the `maxsize` to a set value rather than `UNLIMITED`.
   
   B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.
   
   C. Refer to the Oracle command reference for more information about using the `create tablespace` command.
   ```

**Parent topic:** Setting up a remote Oracle RAC database on Windows for a stand alone production server

**Previous topic:** Setting up databases

**Next topic:** Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:

- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root` /PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

**Prerequisites**

- Installing Oracle RAC
- Create users
- Creating databases
- Modifying database properties
- Setting up databases
- Windows stand-alone server: Creating JCR table spaces

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`
   
   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword `TABLESPACE` and a space. For example: `community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE`
   
   Repeat this step for each domain that you are transferring.
4. Save and close `dbdomain.space_mapping.properties`.
5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - Windows: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
UNIX: `.ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

**Parent topic:** Setting up a remote Oracle RAC database on Windows for a stand alone production server

**Previous topic:** Windows stand-alone server: Creating JCR table spaces

**Next topic:** Configure WebSphere Portal to use Oracle RAC
**Configure WebSphere Portal to use Oracle RAC**

View information on manually transferring data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

*Ensure that the following prerequisites are met:*

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

**Prerequisites**

- Installing Oracle RAC
- Create users
- Creating databases
- Modifying database properties
- Setting up databases
- Windows stand-alone server: Creating JCR table spaces

**Tips:**

- If you are transferring from Oracle, the `open_cursor` setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

- The WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following: `jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME`. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.

- Manually specify an RAC datasource URL in the WebSphere Application Server console. The URL should look like the following:

  `jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT=1521)) (ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=DATABASE SERVICENAME)))`.

- When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

**Steps for transferring data to another supported database**

1. Open a command prompt and change to the directory `wp_profile_root\ConfigEngine`. 
2. Enter the following commands to validate configuration properties.
   ```bash
   ConfigEngine.bat validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ConfigEngine.bat validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ```

3. From the same command prompt as the previous steps, change to the directory `wp_profile_root\bin`.

4. Stop both WebSphere Application Server and the WebSphere Portal server:
   ```bash
   Option                        Description
   WebSphere Application Server  stopServer.bat server1 -username admin_userid -password admin_password
   WebSphere Portal              stopServer.bat WebSphere_Portal -username admin_userid -password admin_password
   ```

5. Transfer the database:
   ```bash
   Option                        Description
   WebSphere Application Server  ConfigEngine.bat database-transfer -DWasPassword=password
   WebSphere Portal              ConfigEngine.bat database-transfer -DTransferDomainList=jcr -DWasPassword=password
   ```
   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example:
   ```sql
   SQL> execute dbms_stats.gather_schema_stats(ownname=>'jcr', cascade=>TRUE);
   ```

7. Specify the JDBC URL to connect to the cluster:
   ```sql
  jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICENAME))
   ```
   **Note:** To select specific database domains to transfer, modify the -DTransferDomainList specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:
   ```bash
   ConfigEngine.bat database-transfer -DTransferDomainList=jcr -DWasPassword=password
   ```

8. Change to the directory `wp_profile_root\bin`.
9. Enter the following command to start the WebSphere Portal server:
   ```bash
   startServer.bat WebSphere_Portal
   ```

**Parent topic:** Setting up a remote Oracle RAC database on Windows for a stand alone production server

**Previous topic:** Assigning custom table spaces
Setting up a remote SQL Server database on Windows for a stand-alone production server

To setup a remote Microsoft SQL Server Enterprise Edition database, create user IDs and databases on a remote server. Tasks are provided to assist with creating the user IDs and the databases. Before you can use the tasks, you must modify properties files.

1. Installing SQL Server
   View the steps to install SQL Server for use with WebSphere Portal.

2. Modifying database properties
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

3. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

   The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

5. Configuring WebSphere Portal to use SQL Server 2005
   View information on manually transferring data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Setting up a database
Installing SQL Server

View the steps to install SQL Server for use with WebSphere Portal.

Before you begin this task, complete the following prerequisites:

- You should have completed reviewing the Planning for SQL Server topic.
- You must install SQL Server separately from WebSphere Portal.
- WebSphere Portal requires either the DataDirect Connect for JDBC 3.7 JDBC driver or the Microsoft SQL Server JDBC Driver.
- You can obtain the DataDirect Driver from DataDirect Technologies. The JDBC driver consists of two parts: the JDBC driver libraries and JTA libraries. The JDBC driver libraries must be put on the WebSphere Portal system, while the JTA libraries are intended for SQL Server.
- You can obtain the Microsoft SQL Server JDBC Driver from Microsoft. See the Microsoft SQL Server product documentation for installation details.

The following section provides instructions for installing SQL Server for use with IBM® WebSphere® Application Server and WebSphere Portal. These steps are the same for both the DataDirect and Microsoft drivers unless noted.

1. Install SQL Server and all required patches.
2. Select the authentication mode for this installation:
   - Windows Authentication Mode
   - Mixed Mode (Windows Authentication and SQL Server Authentication)
     Either authentication mode allows the user who is specified as the database administrator in the wkplc_comp.properties file to log in to the SQL server. Mixed Mode authentication allows either a Windows user or a SQL server user, or both, to log in to the SQL Server.
3. In the SQL Server Setup panel, **Components to Install**, select the following components, which are required services for WebSphere Portal:
   - **SQL Server Database Services**
     - **Integration Services** The option Integration Services, creates the directory DTS/Binn, where you will need to copy files when you install Microsoft SQL Server JDBC drivers and enable XA connections.
4. Complete the installation using SQL Server documentation as a guide.
5. Enable TCP/IP connectivity in the **SQL Server Configuration Manager**.
6. Install the JDBC driver using one of these methods:
   - Installing DataDirect Connect for JDBC drivers on UNIX
   - Installing DataDirect Connect for JDBC drivers on Windows
   - Installing Microsoft SQL Server JDBC drivers and enabling XA connections
7. Network the connections for MSDTC in the Windows Component Services as described in Microsoft Help and Support.
8. Start SQL Server.

**Installing DataDirect Connect for JDBC drivers on UNIX**

1. Purchase and download DataDirect Connect for JDBC and save file 360connectjdbc.jar in a temporary work directory.
2. To create the required files, run the following command from the directory that contains 360connectjdbc.jar:

```
jar -xvf 360connectjdbc.jar
```
3. Run `./Installer.sh` in the same directory.
4. When installation is complete, change directory to `/installation/lib`, where `installation` is the directory in which you installed the DataDirect Connect for JDBC drivers.
5. Run the following command to change the permissions on the JAR files in the `installation/lib` directory:

```
chmod 777 *.jar
```
6. Change the ownership and group of these files by running the following commands in the `installation/lib` directory:

```
chgrp system_grp *.jar
chown root *.jar
```
Where `system_grp` is the system group as labeled by your operating system.

### Installing DataDirect Connect for JDBC drivers on Windows
1. Purchase and download DataDirect Connect for JDBC and save file `360connectjdbc.jar` in a temporary work directory.
2. To create the required files, run the following command from the directory that contains 360connectjdbc.jar:

```
jar -xvf 360connectjdbc.jar
```
3. Run `Installer.bat` in the same directory.

### Installing Microsoft SQL Server JDBC drivers and enabling XA connections
1. Download and install the Microsoft SQL Server JDBC driver.
2. Copy file `sqljdbc_xa.dll` from the `xa` subdirectory to the `C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqljdbc_xa.dll` directory of the SQL Server installation.
3. Start the database server.
4. Ensure that the Distributed Transaction Coordinator has been started. The status can be verified in the list of services in the Computer Management console.
5. Start the Microsoft SQL Server Management Studio and connect to the local database engine as the system administrator, `sa`.
6. Select `File > Open > File` and select `xa_install.sql` from the subdirectory of the downloaded and extracted JDBC driver.
7. Execute the script by selecting `Query > Execute.Note: Any warnings that appear in the messages section of the application window that say that stored procedures cannot be found can be safely ignored.
8. For Microsoft SQL Server JDBC drivers: If you are running Windows XP SP2, Windows XP 64-Bit Edition, or Windows Server 2003, refer to the Registry Entries Are Required for XA Transaction Support document for information on a new security constraint and how to set SQL Server on Windows XP SP2, Windows XP 64-Bit Edition, or Windows Server 2003. Create a additional value in the Windows registry for WebSphere Portal by following these steps:
   A. Open the Windows Registry Editor (regedit) and navigate to the element
      
      ```
      HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSDTC\XADLL
      ```
   B. From the menu bar, select `Edit > New > String Value` to create a new parameter named `sqljdbc_xa.dll` in that element.
   C. Change the value of the new parameter to the location of the `sqljdbc_xa.dll` file copied in Step 2 above, for example:
      
      ```
      C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqljdbc_xa.dll
      ```

### Enable XA Transactions In Windows Component Services
1. From your Windows desktop, follow these steps:
2. Click `Start > Settings > Administrative Tools > Component Services.`
3. Expand the tree view to locate the computer where you want to turn on support for XA transactions (for example, My Computer).
4. Display the context menu for the computer name and click Properties.
5. Click Options and tune the Transaction Timeout that suits your environment. (The recommended minimum is 180 seconds).
6. Click MSDTC and click Security Configuration.
7. Under Security Settings, select XA Transactions to enable this support.
8. Click OK to save your changes.

**Note:** The installation documentation for JDBC XA connectivity refers to two known problems, see the Microsoft support site for more information:
- KB318818: Performance slows down when you use XA Transactions with SQL Server.

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server

**Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - **Installing SQL Server**

  The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

  **Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
  - WebSphere Portal 6.1.5 wkplc.properties file reference
  - WebSphere Portal 6.1.5 wkplc_comp.properties file reference
  - WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**

- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:
  - `release.DbName=release`
  - `jcr.DbName=jcrdb`
  - `feedback.DbName=fdbkdb`
  - `likeminds.DbName=lmdb`
  - `community.DbName=commdb`
  - `customization.DbName=custdb`

- If you are using a remote database, enter the values for the remote server.

- Use a forward slash (/) instead of a backslash (\).

- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.

- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.

- Depending on which database domain has to be configured, replace `dbdomain` with:
  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`

- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
  - `dbdomain.DbType`
  - `dbdomain.DbName`
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.
   A. For dbdomainDbType, type sqlserver2005.
   B. For dbdomainDbName, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the dbdomainDbUrl property.
   C. For dbdomainDbSchema, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   D. For dbdomainDataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For dbdomainDbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:** The database element of this value should match the value of DbName.
   F. For dbdomainDbUser, type the user ID for the database administrator.
   G. For dbdomainDbPassword, type the password for the database administrator.
   H. For dbdomainDBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.
   I. For dbdomainDBA.DbPassword, type the database administrator password for privileged access operations during creation of the database.
   J. For dbdomainDbHome, type the root location for the database. **Note:**
      - This value is the location to store the database files locally.
      - This path must use two backslashes (\) instead of a forward slash (/).
   K. For dbdomainAdminUrl, type the sqlserver URL without a database attached. This value is used to connect to the server for database administration operations.
   L. For dbdomainDbHostName, type the hostname of the database.
3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For `source.domain.DbType`, type of the database you are currently configured to use. The value for `source.domain.DbType` is Derby by default.
B. For `source.domain.DbName`, type the name of the database domain you are currently using.
C. For `source.domain.DbSchema`, type current schema identifier for objects within the database for this domain.
D. For `source.domain.DataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.
E. For `source.domain.DbUrl`, type the url currently used to access your database.
F. For `source.domain.DbUser`, type the name of the user accessing this database.
G. For `source.domain.DbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

A. For `sqlserver2005.DbDriver`, type the name of the class that SqlProcessor uses to import SQL files.
B. For `sqlserver2005.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.
C. For `sqlserver2005.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.
D. For `sqlserver2005.DbConnectionPoolDataSource`, type the name of the implementation class of the connection pool data source.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server  
**Previous topic:** Installing SQL Server  
**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

Before you begin, ensure that the following prerequisites are met:
- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

- **Prerequisites**
  - Installing SQL Server
  - Modifying database properties

1. Change to the directory `wp_profile_root\ConfigEngine`
2. To create the databases, type the following command:
   ```
   ConfigEngine.bat create-database -DWasPassword=password
   ```
3. To create the database users, type the following command:
   ```
   ConfigEngine.bat setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:
   ```
   \<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql
   ```

- **Creating users manually**

Use this alternative method for creating users if you have problems running the setup-database task that is documented for setting up a remote SQL Server 2005 database on Windows for a stand-alone production server.

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server

**Previous topic:** Modifying database properties

**Next topic:** Windows stand-alone server: Assigning custom filegroups on SQL Server 2005
Creating users manually

Use this alternative method for creating users if you have problems running the setup-database task that is documented for setting up a remote SQL Server 2005 database on Windows for a stand-alone production server.

Before you begin:

- You should have completed installing SQL Server 2005 and creating databases.

- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

Create the recommended database users with the SQL Server Management Studio. At least one user is required for each SQL Server 2005 instance.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Connect to your SQL Server 2005 instance.
2. Expand the tree view beneath the SQL Server instance.
3. Expand Security and right-click on Logins.
4. In the opening context menu, select New Login...
5. Enter the database user names.
7. Set a password for the selected user.
8. In the Database Access window, select the database the user must connect to at runtime. The following mappings are recommended: Table 1. List of mappings by database

<table>
<thead>
<tr>
<th>Database</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELEASE</td>
<td>RELEASEUSR</td>
</tr>
<tr>
<td>COMMUNITY</td>
<td>COMMUNITYUSR</td>
</tr>
<tr>
<td>CUSTOMIZATION</td>
<td>CUSTOMIZATIONUSR</td>
</tr>
<tr>
<td>WORKFLOW</td>
<td>WORKFLOWUSR</td>
</tr>
<tr>
<td>WMM</td>
<td>WMMDBUSR</td>
</tr>
<tr>
<td>JCRDB</td>
<td>JCR</td>
</tr>
<tr>
<td>FDBKDB</td>
<td>FEEDBACK</td>
</tr>
<tr>
<td>LMDB</td>
<td>LIKEMINDS</td>
</tr>
</tbody>
</table>

9. Click OK to save the user changes.
10. Add the role SqlJDBCXAUser to all the database users that connect to the database using XA connections. This role is available for the database "master" only. To grant users this role:
   A. Open the Microsoft SQL Server Management Studio and connect to the database instance.
   B. Expand Security > Logins underneath the database instance name.
   C. Click the user name to open the Login Properties window.
   D. Select User Mapping.
E. Select database master in the database list.
F. Select SqlJDBCXAUser.
G. Click OK.

Parent topic: Setting up databases
Windows stand-alone server: Assigning custom filegroups on SQL Server 2005

The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

Before you begin:
- The custom filegroups must exist prior to the execution of the database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating filegroups refer to the documentation for the database.

**Prerequisites**
- Installing SQL Server
- Modifying database properties
- Setting up databases

If custom filegroups are assigned, each must be assigned explicitly. The default filegroups can be used to contain database objects; however the name of the default file group must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom filegroups:

1. Determine the names of your custom filegroups.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tablespaces that specifies the table space and index space for each property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.indexspace.indexspace`
   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
     - release
     - community
     - customization
     - jcr
     - feedback
     - likeminds
3. Assign a filespace to each entry in the mapping file. The filegroup name must be prepended by the keyword `ON` and a space. For example: `community.COMP_INST.tablespace=ON COMM8KSPACE`
   Repeat this step for each domain that you are transferring.
4. Save and close `dbdomain.space_mapping.properties`.
5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - Windows: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
   - UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

**Parent topic:** Setting up a remote SQL Server database on Windows for a stand-alone production server  
**Previous topic:** Setting up databases  
**Next topic:** Configuring WebSphere Portal to use SQL Server 2005
Configuring WebSphere Portal to use SQL Server 2005

View information on manually transferring data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

**- Prerequisites**

- Installing SQL Server
- Modifying database properties
- Setting up databases

**Tips:**

- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

**Steps for transferring data to another supported database**

1. Open a command prompt and change to the directory `wp_profile_root\ConfigEngine`.
2. Enter the following commands to validate configuration properties.
   ```
   ConfigEngine.bat validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ConfigEngine.bat validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```
3. From the same command prompt as the previous steps, change to the directory `wp_profile_root\bin`.
4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td>stopServer.bat server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td>stopServer.bat WebSphere_Portal -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

5. Transfer the database **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

A. Change to the directory `wp_profile_root\ConfigEngine`. 
B. Enter the following command:

```
ConfigEngine.bat database-transfer -DWasPassword=password
```

**Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:

```
ConfigEngine.bat database-transfer -DTransferDomainList=jcr -DWasPassword=password
```

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. Change to the directory `wp_profile_root\bin`.

7. Enter the following command to start the WebSphere Portal server:

```
startServer.bat WebSphere_Portal
```

8. Update the SQL Server 2005 statistics for Portal, and JCR databases by opening SQL Server Management Studio, selecting **New Query**, and running the following query:

```
use db_name
exec sp_updatestats @resample='resample';
```

---

**Parent topic:** [Setting up a remote SQL Server database on Windows for a stand-alone production server](#)

**Previous topic:** [Windows stand-alone server: Assigning custom filegroups on SQL Server 2005](#)
Verifying databases

After you configure IBM® WebSphere® Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.

You can verify the database connection using IBM WebSphere Application Server or by opening WebSphere Portal in a browser.

- To verify that the WebSphere Portal application server is running by using WebSphere Application Server, complete these steps:

  1. Open the WebSphere Application Server administrative console by entering the following address in a browser:

     http://hostname.example.com:10027/ibm/console

     where hostname.example.com is the fully qualified host name of the machine where WebSphere Portal is running and 10027 is the default transport port that is created by WebSphere Application Server.

  2. Log into the administrative console.

  3. Depending on your version of WebSphere Application Server, click the appropriate option:

     - For WebSphere Application Server Version 6.1: Click Resources & JDBC Providers.
     - For WebSphere Application Server Version 7.0: Click Resources > JDBC > JDBC Providers

  4. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.

  5. Select the name of the data source that is defined in wkplc_comp.properties. The default data source is wpdbDS.

  6. Select the name of the JDBC provider that is specified in wkplc_dbtype.properties. The default JDBC provider is wpdbJDBC_dbtype, where dbtype is replaced by the value that matches your environment.

  7. Click Test Connection to verify the database connection. If configuration parameters have been changed, you might need to restart WebSphere Application Server for the test to complete.

- To verify that the WebSphere Portal application server is running by opening WebSphere Portal in a browser, enter the following URL in a supported browser:

http://hostname.example.com:10040/wps/portal

where hostname.example.com is the fully qualified host name of the machine where WebSphere Portal is running and 10040 is the default transport port that is created by WebSphere Application Server.

Parent topic: Setting up a database
Preparing a remote Web server when portal is installed on Windows

Install and configure the Web server plug-in, provided by IBM® WebSphere® Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

- Prerequisites
  - Setting up a database

Perform the following steps to install and configure your Web server:

1. Install and configure the Web server; refer to the Web server documentation for information.

2. If using Microsoft Internet Information Server, we recommend updating the *UrlSegmentMaxLength* Registry key to a value of 0 to eliminate potential problems in a WebSphere Portal environment with the default IIS limitation on the length of URL path segments. We also recommend updating the *AllowRestrictedChars* Registry key to a value of 1 to accept hex-escaped characters in request URLs that decode to the U+0000 - U+001F and U+007F - U+009F ranges. Note: Refer to Http.sys registry settings for IIS for information.

3. If using IBM Lotus® Domino®, edit the *NOTES.INI* file on the Web server. Set the *HTTPEnableConnectorHeaders* parameter to 1.

4. If using IBM HTTP Server or Apache Server, edit the *httpd.conf* file on the Web server. Set the *AllowEncodedSlashes* directive to on; the directive should be added at the root level as a global directive.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>See the appropriate HTTP Server documentation</td>
<td>IBM HTTP Server</td>
</tr>
<tr>
<td>See the appropriate Apache Server documentation</td>
<td>AllowEncodedSlashes directives</td>
</tr>
</tbody>
</table>

5. Stop the Web server.

6. Install and configure the Web server plug-in on the system where the Web server is located using the Plug-ins installation wizard provided with WebSphere Application Server. Refer to Selecting a Web server topology diagram and roadmap for information. Note: The WebSphere Application Server Web server instructions are for the default profile. If not using the default profile, refer to Configuring a Web server for a non-default profile.

- If using WebDAV: After successfully installing the Web server plug-in, locate and open your *plugin-cfg.xml* file and set *AcceptAllContent* to true.

  Important: Depending on how you use the Web server, you may need to adjust the *ServerIOTimeout* value, which defines how long the plug-in should wait for a response from the application. The recommended minimum value is 60 but you may need to adjust this value higher if you are retrieving data from a database. To update this value, locate and open your *plugin-cfg.xml* file and set *ServerIOTimeout* to a value that is appropriate for your business needs. For additional information, see Common questions about the Web server plug-in.

7. If you are using Sun Web Server Version 7 update 8, read Technote 1448262 and perform the steps to resolve the HTTP 408/409 error.

8. Start the Web server.

Parent topic: Setting up a stand-alone server on Windows
Previous topic: Setting up a database
Next topic: Configuring WebSphere Portal to use a user registry on Windows
Configuring WebSphere Portal to use a user registry on Windows

Configure user registry security on IBM® WebSphere® Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

Prior to configuring security, you should use the IBM WebSphere Application Server backupConfig task to create and store a backup of the IBM WebSphere Portal configuration; see backupConfig command for information.

- Prerequisites
  - Setting up a database

Perform the following tasks to configure WebSphere Portal to use a user registry:

1. Preparing user registries on Windows
   Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

2. Choosing your user registry model on Windows
   Choose between securing IBM WebSphere Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

3. Adapting the attribute configuration
   After installing IBM WebSphere Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

4. Stand-alone server: Configuring WebSphere Portal to use dynamic groups on Windows
   By default, WebSphere Portal is enabled for static groups. However, the Virtual Member Manager (VMM) allows users to be members of either static or dynamic groups. Static groups are those where a persistent binding exists between a group and its members. Dynamic groups are those where a search query is defined to retrieve the members of a group. If you have your LDAP server configured to use dynamic groups, complete the steps in this task for WebSphere Portal to use dynamic group queries when you setup your LDAP server.

5. Stand-alone server: Enabling referrals for your LDAP user registry on Windows
   Referrals redirect object requests from one LDAP server to another when objects do not exist or cannot be located in a particular directory tree. You should enable referrals if your environment has more than one user registry existing on multiple servers or domains.

Parent topic: Setting up a stand-alone server on Windows
Previous topic: Preparing a remote Web server when portal is installed on Windows
Next topic: Tune your servers

Related tasks
Managing your user registry on Windows
Preparing user registries on Windows

Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

Choose the appropriate LDAP server to install and setup:

- **Preparing an Active Directory server**
  If you plan to use Active Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing an Active Directory-Lightweight-Directory-Services on Windows**
  If you plan to use an Active Directory-Lightweight-Directory-Services as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Domino Directory server**
  If you plan to use a Domino Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a SecureWay Security Server**
  If you plan to use a SecureWay Security Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Novell eDirectory**
  If you plan to use a Novell eDirectory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Sun Java System Directory Server**
  If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Tivoli Directory Server**
  If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

Parent topic: Configuring WebSphere Portal to use a user registry on Windows

Next topic: Choosing your user registry model on Windows
Preparing an Active Directory server

If you plan to use Active Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Active Directory:

1. Perform the following steps to install and configure Active Directory:
   B. Install required Service Packs.
   D. Perform the following steps to install Internet Information Services (IIS), which is required to export server certificates and must be installed before installing Certificate Services:
      1. Open the Control Panel and select Add/Remove Programs.
      2. Choose Add/Remove Windows Components.
      3. Choose the Internet Information Services (IIS) component and then click Next.
   E. Use the following steps to install Certificate Services if you plan on using Active Directory over SSL:
      1. Open the Control Panel and select Add/Remove Programs.
      2. Choose Add/Remove Windows Components.
      3. Select Certificate Services and then click Next.
      4. Select Stand-alone root CA and then click Next. You can also choose other options depends on you needs.
      5. Fill in CA identifying information and then click Next.
      6. Follow the instruction of the Windows Components Wizard. The Windows Server CD is needed.

2. Perform the following steps as a guide to create the WebSphere Portal administrative user:
   A. Create a new user with the Windows administrative tools. Note: There is a 20 character limitation for the user account name.
   B. Set the password for the new user.
   C. Activate the new user with the Windows administrative tools. Set the msDS-UserAccountDisabled attribute to false.

3. Perform the following steps to enable SSL for Active Directory; this step is required to set passwords during sign up and user creation:
   A. Install an Enterprise Certificate Authority on a Windows 2000 Domain Controller, which installs a certificate on a server or install a third-party certificate on the Domain Controller.
   B. Click Start > All Programs > Administrative Tools > Active Directory Users and Computer.
   C. In the Active Directory Users and Computers window, right-click on your domain name and select Properties.
   D. In the Domain Properties dialog box, select the Group Policy tab.
   E. Select the Default Domain Policy group policy and then click Edit.
   F. Select Windows Settings under Computer Configuration.
   G. Select Security Settings and then select Public Key Policies.
   H. Select Automatic Certificate Request Settings.
      1. Use the wizard to add a policy for Domain Controllers. Note: When these requirements are complete, all domain controllers request a certificate and support LDAP over SSL using port 636.
Preparing an Active Directory-Lightweight-Directory-Services on Windows

If you plan to use an Active Directory-Lightweight-Directory-Services as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Active Directory-Lightweight-Directory-Services:


2. Perform the following steps as a guide to create the WebSphere Portal administrative user:
   A. Create a new user with the Windows administrative tools. **Note:** There is a 20 character limitation for the user account name.
   B. Set the password for the new user.
   C. Activate the new user with the Windows administrative tools. Set the msDS-UserAccountDisabled attribute to false.

**Parent topic:** Preparing user registries on Windows
Preparing a Domino Directory server

If you plan to use a Domino Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Domino Directory:

1. Perform the following steps to prepare the Domino Directory:
   A. Go to Lotus Domino documentation.
   B. Select the appropriate version tab for your product.
   C. Click the Download/View online link for the Lotus Domino for multiple platforms Information Center.
   D. Click Domino Administrator Help > Installation > Installing and setting up Domino servers > Server installation > Installing Domino on Windows and perform this task.
   E. Click Domino Administrator Help > Installation > Installing and setting up Domino servers > The Domino server setup program and perform this task.

2. Perform the following steps as a guide to create the WebSphere Portal administrative user:
   A. Navigate to the People view of the Domino Directory and then click Add Person.
   B. Enter the following values in the New Person form to create the wpsbind user:
      - Last Name
        - wpsbind
      - User Name
        - wpsbind/DominoDomain, where DominoDomain is your Lotus Domino Internet domain
        - wpsbind Note: Make sure you enter two values in the User Name field, where the first value includes the Lotus Domino domain.
      - Short name/UserID
        - wpsbind
      - Internet password
        - wpsbind
   C. Click Save and Close to save the new person record for wpsbind and return to the People view.
   D. Click Add Person and enter the following values in the New Person form to create the wpsadmin user:
      - Last Name
        - wpsadmin, where wpsadmin in the user ID for the WebSphere Portal Administrator
      - User Name
        - wpsadmin/DominoDomain, where DominoDomain is your Lotus Domino Internet domain
        - wpsadmin Note: Make sure you enter two values in the User Name field, where the first value includes the Lotus Domino domain.
      - Short name/UserID
        - wpsadmin
      - Internet password
        - wpsadmin
   E. Click Save and Close to save the new person record for wpsadmin and return to the People view.
   F. Navigate to the Groups view and click Add Group.
   G. Enter the following values in the New Group form on the Basic tab:
      - Group name
- **wpsadmins**
  
  **Note:** If you plan to configure WebSphere Portal for multiple user registries and your Lotus Domino LDAP will share a realm with another user registry, you must use the hierarchical naming convention for the group names, for example: wpsadmins/DominoDomain, to avoid unexpected results during WebSphere Portal runtime.

- **Group type**
  - Multi-purpose

- **Members**
  - wpsbind/DominoDomain
  - wpsadmin/DominoDomain

  **Note:** You can add additional administrator users if required.

H. Click **Save and Close** to save the wpsadmins group with the wpsbind and wpsadmin users as members.

3. Perform the following steps to update the access control list for the Domino Directory:

   A. Open the *names.nsf* file in the Lotus Domino Administrator or Lotus Notes client.

   B. Click **File > Application > Access Control** from the main menu to open the access control list for the file.

   C. In the **Access Control List > Basics** panel, ensure that the wpsadmins group has either Author or Editor access.

   D. Add the following Role Types to the wpsadmins group:
      - GroupCreator
      - GroupModifier
      - UserCreator
      - UserModifier

   E. Click **OK**.

**Parent topic:** [Preparing user registries on Windows](#)
Preparing a SecureWay Security Server

If you plan to use a SecureWay Security Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare SecureWay Security Server:

1. Install SecureWay Security Server; refer to the IBM SecureWay™ Security Server for z/OS® and OS/390® for information.

2. Perform the following steps to create the WebSphere Portal administrative user:
   
   **A.** Optional: Perform the following steps to create a new directory suffix:
   
   1. Click the **Server Administration** folder, located in the left-hand navigation of the directory server console.
   2. Click the **Manage Server Properties** folder under the Server Administration folder and then select **Suffixes** on the main page.
   3. Type the **Base DN** name for the suffix; for example: `dc=yourcompany,dc=com`.
   4. Click **Add**.
   5. Click **OK** to save your changes.
   6. Stop and restart the LDAP server.

   **B.** Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:

   - Use the **PortalUsers.ldif** file as a working example and adapted appropriately to work with your LDAP server.
   - Use the **ContentUsers.ldif** file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.

   **C.** Replace every `dc=yourco,dc=com` with your suffix.

   **D.** Replace any prefixes and suffixes that are unique to your LDAP server.

   **E.** You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.

   **F.** Optional: If using IBM Tivoli® Access Manager Version 5.1, set the **objectclasses** to **accessGroup**. If using Tivoli Access Manager Version 6, set the **objectclasses** to **groupOfNames**.

   **G.** Save your changes.

   **H.** Follow the instructions provided with your directory server to import the LDIF file.

   **I.** Stop and restart the LDAP server.

**Parent topic:** Preparing user registries on Windows
Preparing a Novell eDirectory

If you plan to use a Novell eDirectory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Novell eDirectory:
1. Install Novell eDirectory; refer to eDirectory for information.
2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every dc=yourco,dc=com with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.
   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.
Preparing a Sun Java System Directory Server

If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Sun Java System Directory Server:

1. Install Sun Java System Directory Server; refer to Oracle software for information.
2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every dc=yourco,dc=com with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.
   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on Windows
Preparing a Tivoli Directory Server

If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Tivoli Directory Server:

   
   **Note:** Due to a restriction in Tivoli Directory Server, users or groups must not contain a Turkish uppercase dotted I or lowercase dotted i in the DN as this will prevent correct retrieval of that user or group.

2. Perform the following steps to create the WebSphere Portal administrative user:
   
   A. Optional: Perform the following steps to create a new directory suffix:
      
      1. Click the **Server Administration** folder, located in the left-hand navigation of the directory server console.
      2. Click the **Manage Server Properties** folder under the Server Administration folder and then select **Suffixes** on the main page.
      3. Type the **Base DN** name for the suffix; for example: `dc=yourcompany,dc=com`.
      4. Click **Add**.
      5. Click **OK** to save your changes.
      6. Stop and restart the LDAP server.

   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      
      - Use the `PortalUsers.ldif` file as a working example and adapted appropriately to work with your LDAP server.
      - Use the `ContentUsers.ldif` file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.

   C. Replace every `dc=yourco,dc=com` with your suffix.

   D. Replace any prefixes and suffixes that are unique to your LDAP server.

   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.

   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to `accessGroup`. If using Tivoli Access Manager Version 6, set the objectclasses to `groupOfNames`.

   G. Save your changes.

   H. Follow the instructions provided with your directory server to import the LDIF file.

   I. Stop and restart the LDAP server.

**Parent topic:** Preparing user registries on Windows
Choosing your user registry model on Windows

Choose between securing IBM® WebSphere® Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

Choose one of the following user registry models:

- **Configuring a stand-alone LDAP user registry on Windows**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Configuring the default federated repository on Windows**
  Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

Parent topic: Configuring WebSphere Portal to use a user registry on Windows
Previous topic: Preparing user registries on Windows
Next topic: Adapting the attribute configuration
Configuring a stand-alone LDAP user registry on Windows

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Configuring a stand-alone LDAP user registry on Windows**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry to store all user account information for authorization.

- **Configuring a stand-alone LDAP user registry over SSL on Windows**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

**Parent topic:** Choosing your user registry model on Windows
Configuring a stand-alone LDAP user registry on Windows

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry to store all user account information for authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps. If you need to rerun the wp-modify-ldap-security task to change the LDAP repositories or because the task failed, you must choose a new name for the realm using the standalone.ldap.realm parameter or you can set ignoreDuplicateIDs=true in the wkplc.properties file, before rerunning the task.

Perform the following steps to configure a standalone LDAP user registry:

Note: Use the wp_security_xxx.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_security_xxx.properties helper file.

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root\ConfigEngine\properties directory.
2. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM Stand-alone LDAP configuration heading:

   - standalone.ldap.id
   - standalone.ldap.host
   - standalone.ldap.port
   - standalone.ldap.bindDN
   - standalone.ldap.bindPassword
   - standalone.ldap.ldapServerType
   - standalone.ldap.userIdMap
   - standalone.ldap.groupIdMap
   - standalone.ldap.groupMemberIdMap
   - standalone.ldap.userFilter
   - standalone.ldap.groupFilter
   - standalone.ldap.serverId
   - standalone.ldap.serverPassword
   - standalone.ldap.realm
   - standalone.ldap.primaryAdminId
   - standalone.ldap.primaryAdminPassword
   - standalone.ldap.primaryPortalAdminId
   - standalone.ldap.primaryPortalAdminPassword
   - standalone.ldap.primaryPortalAdminGroup
   - standalone.ldap.baseDN
3. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading:
   - `standalone.ldap.et.group.objectClasses`
   - `standalone.ldap.et.group.objectClassesForCreate`
   - `standalone.ldap.et.group.searchBases`
   - `standalone.ldap.et.personaccount.objectClasses`
   - `standalone.ldap.et.personaccount.objectClassesForCreate`
   - `standalone.ldap.et.personaccount.searchBases`

4. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attributes heading:
   - `standalone.ldap.gm.groupMemberName`
   - `standalone.ldap.gm.objectClass`
   - `standalone.ldap.gm.scope`
   - `standalone.ldap.gm.dummyMember`

5. Required: Enter a value for the following required relative distinguished name (RDN®) parameters in the `wkplc.properties` file under the Default parent, RDN attribute heading:
   - `standalone.ldap.personAccountParent`
   - `standalone.ldap.groupParent`
   - `standalone.ldap.personAccountRdnProperties`
   - `standalone.ldap.groupRdnProperties`

6. Save your changes to the `wkplc.properties` file.

7. Optional: Enter the following Web content authors parameters in the `wkplc_comp.properties` file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product:
   - `WcmContentAuthorsGroupId`
   - `WcmContentAuthorsGroupCN`

8. Run the `ConfigEngine.bat validate-standalone-ldap -DWasPassword=password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, `WasPassword` is the value entered during installation and not a value found in your LDAP user registry.
   
   **Note:** During the validation task, you may receive the following prompt: *Add signer to the trust store now?*. Press `y` then `Enter`.

9. Run the `ConfigEngine.bat wp-modify-ldap-security -DWasPassword=password` task, from the `wp_profile_root\ConfigEngine` directory, to set the stand-alone LDAP user registry.

10. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents*.

11. Run the `ConfigEngine.bat wp-validate-standalone-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root\ConfigEngine` directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

12. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.
LDAP directory.

**Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the configure-express task.

A. Edit the `wp_profile_root` PortalServer\wcm\shared\app\config\wcmservices\MemberFixerModule.properties file.

B. Add the following lines to the file:

```
uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
```

Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `ConfigEngine.bat action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root\ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

13. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

**Parent topic:** Configuring a stand-alone LDAP user registry on Windows

**Related tasks**
- Adapting the attribute configuration
- Using the member fixer tool with IBM Lotus Web Content Management
- Starting and stopping servers, deployment managers, and node agents

### Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>
Configuring a stand-alone LDAP user registry over SSL on Windows

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to configure a standalone LDAP user registry over SSL: **Note:** Use the wp_security_xxx.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_security_xxx.properties helper file.

1. To specify the LDAP server’s SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:
   - **Server trust store**
     A. Add the certificate to the trust store:
       1. Log in to the WebSphere Application Server Administrative Console.
       2. Navigate to Security > SSL certificate and key management > SSL configurations.
       3. Click the appropriate SSL configuration from the list. For example,
          - Stand-alone environments: NodeDefaultSSLSettings
          - Clustered environments: CellDefaultSSLSettings

     **Clustered environments:** Ensure the setting for SSL configuration for outbound connection matches your SSL settings.

     4. Click Key stores and certificates.
     5. Click the appropriate trust store from the list; for example, NodeDefaultTrustStore.
     6. Click Signer certificates, click Add, and then enter the following information:
        - Type the Alias the key store uses for the signer certificate.
        - Type the File name where the signer certificate is located.
     7. Click OK and then click Save to save the changes to the master configuration.

   B. Retrieve the certificate from the port:
      1. Log in to the WebSphere Application Server Administrative Console.
      2. Navigate to Security > SSL certificate and key management > SSL configurations.
      3. Click the appropriate SSL configuration from the list. For example,
         - Stand-alone environments: NodeDefaultSSLSettings
Clustered environments: Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.

4. Click **Key stores and certificates**.

5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.

6. Click **Signer certificates**, click **Retrieve from port**, and then enter the following information:
   - Type the **Host** name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the **Port** used when attempting to retrieve the signer certificate.
   - Type the **Alias** the key store uses for the signer certificate.

7. Click **Retrieve signer information** to retrieve the certificate from the port.

8. Click **OK** and then click **Save** to save the changes to the master configuration.

- **Client trust store** **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

  A. See Secure installation for client signer retrieval.

  B. Run the retrieveSigners task from the **wp_profile_root\bin** directory; see retrieveSigners command for information. In a deployed environment, you will need to run the retrieveSigners task, for any federated node, against the Deployment Manager. **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

C. Update the trust store properties file:

1. Use a text editor to open the **ssl.client.props** file, located in the **wp_profile_root\properties** directory.

2. Change the **com.ibm.ssl.trustStore** parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter **com.ibm.ssl.trustStore=${CONFIG_ROOT}/cells/wpsbvt/nodes/wpsbvt/trust.p12** to use the default trust store.

3. Save your changes.

2. Use a text editor to open the **wkplc.properties** file, located in the **wp_profile_root\ConfigEngine\properties** directory.

3. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM Standalone LDAP configuration heading. **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - **standalone.idap.id**
   - **standalone.idap.host**
   - **standalone.idap.port**
   - **standalone.idap.blndDN**
   - **standalone.idap.bindPassword**
   - **standalone.idap.idapServerType**
   - **standalone.idap.userldMap**
   - **standalone.idap.groupldMap**
   - **standalone.idap.groupMemberldMap**
   - **standalone.idap.userFilter**
   - **standalone.idap.groupFilter**
   - **standalone.idap.serverld**
   - **standalone.idap.serverPassword**
   - **standalone.idap.realm**
   - **standalone.idap.primaryAdminld**
   - **standalone.idap.primaryAdminPassword**
   - **standalone.idap.primaryPortalAdminld**
- standalone.ldap.primaryPortalAdminPassword
- standalone.ldap.primaryPortalAdminGroup
- standalone.ldap.baseDN

4. Required: Enter a value for the following required entity types parameters in the wkplc.properties file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.et.group.objectClasses
   - standalone.ldap.et.group.objectClassesForCreate
   - standalone.ldap.et.group.searchBases
   - standalone.ldap.et.personaccount.objectClasses
   - standalone.ldap.et.personaccount.objectClassesForCreate
   - standalone.ldap.et.personaccount.searchBases

5. Required: Enter a value for the following required group member parameters in the wkplc.properties file under the Group member attributes heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.gm.groupMemberName
   - standalone.ldap.gm.objectClass
   - standalone.ldap.gm.scope
   - standalone.ldap.gm.dummyMember

6. Required: Enter a value for the following required relative distinguished name (RDN®) parameters in the wkplc.properties file under the Default parent, RDN attribute heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.personAccountParent
   - standalone.ldap.groupParent
   - standalone.ldap.personAccountRdnProperties
   - standalone.ldap.groupRdnProperties

7. Enter a value for the following parameters to enable Secure Socket Layers (SSL): **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   Required parameters:
   - standalone.ldap.sslEnabled
   - standalone.ldap.sslConfiguration
   Optional parameters:
   - standalone.ldap.certificateMapMode
   - standalone.ldap.certificateFilter

8. Save your changes to the wkplc.properties file.
9. Run the ConfigEngine.bat validate-standalone-ldap -DWasPassword=password task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, **WasPassword** is the value entered during installation and not a value found in your LDAP user registry. **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press y then Enter.
10. Run the ConfigEngine.bat wp-modify-ldap-security -DWasPassword=password task, from the wp_profile_root\ConfigEngine directory, to set the stand-alone LDAP user registry.
11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: **Starting and stopping servers, deployment managers, and node agents.**
12. Run the ConfigEngine.bat wp-validate-standalone-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root\ConfigEngine directory, to check that all defined attributes are available in the configured
LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

**Parent topic:** Configuring a stand-alone LDAP user registry on Windows

**Related tasks**
Adapting the attribute configuration
Starting and stopping servers, deployment managers, and node agents
Configuring the default federated repository on Windows

Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

Perform the following tasks to add user registries to the default federated repository:

- **Configuring a federated LDAP user registry on Windows**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Adding a database user registry on Windows**
  Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

- **Adding realm support on Windows**
  A realm is a group of users from one or more user registries that form a coherent group within IBM WebSphere Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

**Parent topic:** Choosing your user registry model on Windows
Configuring a federated LDAP user registry on Windows

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Adding an LDAP user registry on Windows**
  Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time. If IBM Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=`groupName and the hierarchical format is `cn=`groupName,o=root. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as wpsadmin, this ID cannot exist in the LDAP you are adding.

- **Adding an LDAP user registry over SSL on Windows**
  Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

**Parent topic:** Configuring the default federated repository on Windows
Adding an LDAP user registry on Windows

Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

If IBM® Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=groupName` and the hierarchical format is `cn=groupName,o=root`. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as `wpsadmin`, this ID cannot exist in the LDAP you are adding.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

1. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root\ConfigEngine\properties` directory.
2. Required: Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated LDAP Properties heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.

   - `federated.ldap.id`
   - `federated.ldap.host`
   - `federated.ldap.port`
   - `federated.ldap.bindDN`
   - `federated.ldap.bindPassword`
   - `federated.ldap.ldapServerType`
   - `federated.ldap.baseDN`

3. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.

   - `federated.ldap.et.group.objectClasses`
   - `federated.ldap.et.group.objectClassesForCreate`
   - `federated.ldap.et.group.searchBases`
   - `federated.ldap.et.personaccount.objectClasses`
   - `federated.ldap.et.personaccount.objectClassesForCreate`
   - `federated.ldap.et.personaccount.searchBases`
4. Required: Enter a value for the following required group member parameters in the wkplc.properties file under the Group member attribute heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - federated.ldap.gm.groupMemberName
   - federated.ldap.gm.objectClass
   - federated.ldap.gm.scope
   - federated.ldap.gm.dummyMember

5. Save your changes to the wkplc.properties file.

6. Optional: Enter the following Web content authors parameters in the wkplc_comp.properties file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. **Note:** See the wkplc_comp.properties file for specific information about the required parameters and for advanced parameters.
   - WcmContentAuthorsGroupId
   - WcmContentAuthorsGroupCN

7. Run the `ConfigEngine.bat validate-federated-ldap -DWasPassword=password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, WasPassword is the value entered during installation and not a value found in your LDAP user registry.
   **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press `y` then `Enter`.

8. Run the `ConfigEngine.bat wp-create-ldap -DWasPassword=password` task, from the `wp_profile_root\ConfigEngine` directory, to add an LDAP user registry to the default federated repository. **Note:** Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

10. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:
    A. Use a text editor to open the wkplc.properties file, located in the `wp_profile_root\ConfigEngine\properties` directory.
    B. Enter a value for the following required parameters in the wkplc.properties file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms:
       **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
       - id
       - baseDN
       - nameInRepository
    C. Save your changes to the wkplc.properties file.
    D. Run the `ConfigEngine.bat wp-create-base-entry -DWasPassword=password` task, from the `wp_profile_root\ConfigEngine` directory, to create a base entry in a repository.
    E. Stop and restart all necessary servers to propagate your changes.

11. Optional: Run the `ConfigEngine.bat wp-query-repository -DWasPassword=password` task, from the `wp_profile_root\ConfigEngine` directory, to list the names and types of configured repositories.

12. Run the `ConfigEngine.bat wp-validate-federated-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root\ConfigEngine` directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between
WebSphere Portal and the LDAP server.

13. Perform the following steps to update the user registry where new users and groups are stored:

   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root\ConfigEngine\properties` directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

      **Note:** See the `properties` file for specific information about the required parameters and for advanced parameters.

      - `personAccountParent`
      - `groupParent`
      - `personAccountRdnProperties`
      - `groupRdnProperties`

      The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:

      - `personAccountParent=dc=yourco,dc=com`
      - `groupParent=dc=yourco,dc=com`

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `ConfigEngine.bat wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root\ConfigEngine` directory, to delete the old attributes before adding the new attributes.

   E. Stop and restart all necessary servers to propagate your changes.

14. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root\ConfigEngine\properties` directory.

   B. Enter a value for `realmName` or leave blank to update the default realm.

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `ConfigEngine.bat wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root\ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `ConfigEngine.bat wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

   E. Stop and restart all necessary servers to propagate your changes.

15. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

   **Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

   A. Edit the `wp_profile_root\PortalServer\wcm\shared\app\config\wcmservices\MemberFixerModule.properties` file.

   B. Add the following lines to the file:

   ```properties
   uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
   cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
   ```

   Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.
C. Save your changes and close the file.

D. Run the `ConfigEngine.bat action-express-memberfixer -DmemberfixerRealm=realm_name` -
`DPortalAdminPwd=password` -`DWasPassword=password` task, located in the `wp_profile_root\ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. **Value for `realm_name`** when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.idap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

16. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

17. Optional: **This step is required in a production environment.** Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

   **Important:**
   - Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
   - Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

   **Note:** If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

   A. Run the following task from the `wp_profile_root\ConfigEngine` directory: `ConfigEngine.bat wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword` **Important:** You must provide the full distinguished name (DN) for the `newAdminid` parameter.

   B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

   C. Run the following task from the `wp_profile_root\ConfigEngine` directory: `ConfigEngine.bat wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupid=newadminGroupid` **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

   D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

18. Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it.

   The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by
duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: *Deleting the repository.*

**Parent topic:** Configuring a federated LDAP user registry on Windows

**Related tasks**
Adapting the attribute configuration
Deleting the repository on Windows
Starting and stopping servers, deployment managers, and node agents

**Related information**
User IDs and passwords
Adding an LDAP user registry over SSL on Windows

Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

In single server environments, you do not have to start or stop the WebSphere Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry over SSL to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

**Note:** Use the `wp_add_federated_xxx.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_add_federated_xxx.properties` helper file.

1. To specify the LDAP server’s SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:
   - **Server trust store**
     A. Add the certificate to the trust store:
        1. Log in to the WebSphere Application Server Administrative Console.
        2. Navigate to Security > SSL certificate and key management > SSL configurations.
        3. Click the appropriate SSL configuration from the list. For example,
           - Stand-alone environments: `NodeDefaultSSLSettings`
           - Clustered environments: `CellDefaultSSLSettings`
           **Clustered environments:** Ensure the setting for SSL configuration for outbound connection matches your SSL settings.
        4. Click **Key stores and certificates**.
        5. Click the appropriate trust store from the list; for example, `NodeDefaultTrustStore`.
        6. Click **Signer certificates**, click **Add**, and then enter the following information:
           - Type the **Alias** the key store uses for the signer certificate.
           - Type the **File name** where the signer certificate is located.
        7. Click **OK** and then click **Save** to save the changes to the master configuration.
   B. Retrieve the certificate from the port:
      1. Log in to the WebSphere Application Server Administrative Console.
      2. Navigate to Security > SSL certificate and key management > SSL configurations.
      3. Click the appropriate SSL configuration from the list. For example,
         - Stand-alone environments: `NodeDefaultSSLSettings`
Clustered environments: CellDefaultSSLSettings

Clustered environments: Ensure the setting for SSL configuration for outbound connection matches your SSL settings.

4. Click Key stores and certificates.

5. Click the appropriate trust store from the list; for example, NodeDefaultTrustStore.

6. Click Signer certificates, click Retrieve from port, and then enter the following information:
   - Type the Host name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the SSL Port used when attempting to retrieve the signer certificate.
   - Type the Alias the key store uses for the signer certificate.

7. Click Retrieve signer information to retrieve the certificate from the port.

8. Click OK and then click Save to save the changes to the master configuration.

- Client trust store Note: This task might report an error, but it does successfully update the trust store. You can ignore the error message.

A. See Secure installation for client signer retrieval.

B. Run the retrieveSigners task from the wp_profile_root\bin directory; see retrieveSigners command for information. In a deployed environment, you will need to run the retrieveSigners task, for any federated node, against the Deployment Manager. Note: This task might report an error, but it does successfully update the trust store. You can ignore the error message.

C. Update the trust store properties file:

1. Use a text editor to open the ssl.client.props file, located in the wp_profile_root\properties directory.

2. Change the com.ibm.ssl.trustStore parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter com.ibm.ssl.trustStore=${CONFIG_ROOT}/cells/wpsbvt/nodes/wpsbvt/trust.p12 to use the default trust store.

3. Save your changes.

2. Use a text editor to open the wkplc.properties file, located in the wp_profile_root\ConfigEngine\properties directory.

3. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM Federated LDAP Properties heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.

   - federated.ldap.id
   - federated.ldap.host
   - federated.ldap.port
   - federated.ldap.bindDN
   - federated.ldap.bindPassword
   - federated.ldap.ldapServerType
   - federated.ldap.baseDN

4. Required: Enter a value for the following required entity types parameters in the wkplc.properties file under the LDAP entity types heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.

   - federated.ldap.et.group.objectClasses
   - federated.ldap.et.group.objectClassesForCreate
   - federated.ldap.et.group.searchBases
   - federated.ldap.et.personaccount.objectClasses
   - federated.ldap.et.personaccount.objectClassesForCreate
   - federated.ldap.et.personaccount.searchBases
5. Required: Enter a value for the following required group member parameters in the \texttt{wkplc.properties} file under the Group member attribute heading: \textbf{Note:} See the properties file for specific information about the required parameters and for advanced parameters.

- \texttt{federated.idap.gm.groupMemberName}
- \texttt{federated.idap.gm.objectClass}
- \texttt{federated.idap.gm.scope}
- \texttt{federated.idap.gm.dummyMember}

6. Enter a value for the following parameters to enable Secure Socket Layers (SSL): \textbf{Note:} See the properties file for specific information about the required parameters and for advanced parameters.

Required parameters:

- \texttt{federated.idap.sslEnabled}
- \texttt{federated.idap.sslConfiguration}

Optional parameters:

- \texttt{federated.idap.certificateMapMode}
- \texttt{federated.idap.certificateFilter}

7. Save your changes to the \texttt{wkplc.properties} file.

8. Optional: Enter the following Web content authors parameters in the \texttt{wkplc_comp.properties} file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. \textbf{Note:} See the \texttt{wkplc_comp.properties} file for specific information about the required parameters and for advanced parameters.

- \texttt{WcmContentAuthorsGroupId}
- \texttt{WcmContentAuthorsGroupCN}

9. Run the \texttt{ConfigEngine.bat validate-federated-ldap -DWasPassword=password} task to validate your LDAP server settings. \textbf{Attention:} If you have not deleted the default file repository, \texttt{WasPassword} is the value entered during installation and not a value found in your LDAP user registry.

\textbf{Note:} During the validation task, you may receive the following prompt: \texttt{Add signer to the trust store now?}. Press \texttt{y} then \texttt{Enter}.

10. Run the \texttt{ConfigEngine.bat wp-create-ldap -DWasPassword=password} task, from the \texttt{wp_profile_root\ConfigEngine} directory, to add an LDAP user registry to the default federated repository. \textbf{Note:} Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: \textit{Starting and stopping servers, deployment managers, and node agents}.

12. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:

A. Use a text editor to open the \texttt{wkplc.properties} file, located in the \texttt{wp_profile_root\ConfigEngine\properties} directory.

B. Enter a value for the following required parameters in the \texttt{wkplc.properties} file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms:

\textbf{Note:} See the properties file for specific information about the required parameters and for advanced parameters.

- \texttt{id}
- \texttt{baseDN}
- `nameInRepository`
C. Save your changes to the `wkplc.properties` file.
D. Run the `ConfigEngine.bat wp-create-base-entry -DWasPassword=password` task, from the `wp_profile_root\ConfigEngine` directory, to create a base entry in a repository.

E. Stop and restart all necessary servers to propagate your changes.

13. Optional: Run the `ConfigEngine.bat wp-query-repository -DWasPassword=password` task, from the `wp_profile_root\ConfigEngine` directory, to list the names and types of configured repositories.

14. Run the `ConfigEngine.bat wp-validate-federated-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root\ConfigEngine` directory, to check that all defined attributes are available in the configured LDAP user registry. Important: When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

15. Perform the following steps to update the user registry where new users and groups are stored:
   - **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.
   - **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root\ConfigEngine\properties` directory.

B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:
   - **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `personAccountParent`
   - `groupParent`
   - `personAccountRdnProperties`
   - `groupRdnProperties`
   - The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:
     - `personAccountParent=dc=yourco,dc=com`
     - `groupParent=dc=yourco,dc=com`

   C. Save your changes to the `wkplc.properties` file.

D. Run the `ConfigEngine.bat wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root\ConfigEngine` directory, to delete the old attributes before adding the new attributes.

E. Stop and restart all necessary servers to propagate your changes.

16. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root\ConfigEngine\properties` directory.

B. Enter a value for `realmName` or leave blank to update the default realm.

C. Save your changes to the `wkplc.properties` file.

D. Run the `ConfigEngine.bat wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root\ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `ConfigEngine.bat wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

E. Stop and restart all necessary servers to propagate your changes.

17. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the
Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

**Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

A. Edit the `wp_profile_root`\App\config\wcmservices\MemberFixerModule.properties file.

B. Add the following lines to the file:

```
uid=xyzadmin, o=defaultWIMFileBasedRealm -> portal_admin_DN
cn=contentauthors, o=defaultWIMFileBasedRealm -> content_authors_group_DN
```

Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `ConfigEngine.bat` action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password task, located in the `wp_profile_root\ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the wkplc.properties file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the wkplc.properties file. If the value for <code>federated.realm</code> is empty, use defaultWIMFileBasedRealm as the default value.</td>
</tr>
</tbody>
</table>

18. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

19. Optional: **This step is required in a production environment.** Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

**Important:**
- Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
- Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

**Note:** If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

A. Run the following task from the `wp_profile_root\ConfigEngine` directory: `ConfigEngine.bat` wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword **Important:** You must provide the full distinguished name (DN) for the `newAdminid` parameter.

B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.
C. Run the following task from the *wp_profile_root*\ConfigEngine directory: `ConfigEngine.bat wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid`. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

20. Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it. The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: Deleting the repository.

**Parent topic:** Configuring a federated LDAP user registry on Windows

**Related tasks**
- Adapting the attribute configuration
- Deleting the repository on Windows
- Starting and stopping servers, deployment managers, and node agents

**Related information**
- User IDs and passwords
Adding a database user registry on Windows

Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add a database user registry to the default federated repository; you must repeat these steps for each additional database user registry that you plan to add. Note: Use the wp_add_DB.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_add_DB.properties helper file.

1. Set up a new database, including creating a new user with appropriate database privileges for accessing the database:

Instructions for setting up databases: Refer to the appropriate documentation for the type of database you want to set up.

Consulting your database administrator: The task of setting up a new database is typically performed by a database administrator. However, the following steps are provided for your reference in the event you are creating a stand-alone database for testing or demonstration purposes. Consult your database administrator before proceeding with the following steps if you plan to create a database for a production environment.

Table 1. Steps for creating a new database to use as a database user registry.

<table>
<thead>
<tr>
<th>Database</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td>Perform the following steps to create a DB2 database: Install DB2. Enter the following database tuning commands: &lt;br&gt; <code>db2 CREATE DB dbname using codeset UTF-8</code> &lt;br&gt; <code>db2 UPDATE DB CFG FOR dbname USING applheapsz 4096</code> &lt;br&gt; <code>db2 UPDATE DB CFG FOR dbname USING app_ctl_heap_sz 1024</code> &lt;br&gt; <code>db2 UPDATE DB CFG FOR dbname USING stmtheap 32768</code> &lt;br&gt; <code>db2 UPDATE DB CFG FOR dbname USING dbheap 2400</code> &lt;br&gt; <code>db2 UPDATE DB CFG FOR dbname USING locklist 1000</code> &lt;br&gt; <code>db2 UPDATE DB CFG FOR dbname USING logfilsiz 4000</code> &lt;br&gt; <code>db2 UPDATE DB CFG FOR dbname USING logprimary 12</code> &lt;br&gt; <code>db2 UPDATE DB CFG FOR dbname USING logsecond 20</code> &lt;br&gt; <code>db2 UPDATE DB CFG FOR dbname USING logbufsz 32</code> &lt;br&gt; <code>db2 UPDATE DB CFG FOR dbname USING avg_appls 5</code> &lt;br&gt; <code>db2 UPDATE DB CFG FOR dbname USING locktimeout 30</code> &lt;br&gt; <code>db2 UPDATE DB CFG FOR dbname using AUTO_MAINT off</code></td>
</tr>
</tbody>
</table>
Perform the following steps to define the **DbDriver** and **DbLibrary** parameter values:

A. Use a text editor to open the `wkplc_dbtype.properties` file, located in the `wp_profile_root\ConfigEngine\properties` directory.

B. Enter a value for the following parameters under the appropriate database type properties heading:
   - `db_type.DbDriver`
   - `db_type.DbLibrary`

C. Save your changes.

**Limitation:** The WebSphere Application Server UserManagement component (VMM) requires access to the following database libraries to use the VMM database functions such as Property Extension and database user registry, however, if the Portal is using the DB2 Type 2 driver, due to functional limitations, VMM must use the DB2 Type 4 driver; see **Configuring a JDBC provider and datasource for federated repositories** for additional information:

<table>
<thead>
<tr>
<th>Database</th>
<th>Oracle</th>
<th>SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Type 2 driver</td>
<td><code>db2java.zip</code></td>
<td><code>db2jcc.jar;db2jcc_license_cu.jar</code></td>
</tr>
<tr>
<td>DB2 Type 4 driver</td>
<td><code>db2jcc.jar;db2jcc_license_cu.jar</code></td>
<td><code>db2jcc.jar;db2jcc_license_cu.jar;db2jcc_javax.jar</code></td>
</tr>
<tr>
<td>DB2 for z/OS Type 2 driver</td>
<td><code>db2java.zip</code></td>
<td><code>db2jcc.jar;db2jcc_license_cisuz.jar;db2jcc_javax.jar</code></td>
</tr>
<tr>
<td>DB2 for z/OS Type 4 driver</td>
<td><code>db2jcc.jar;db2jcc_license_cisuz.jar;db2jcc_javax.jar</code></td>
<td><code>db2jcc.jar;db2jcc_license_cisuz.jar;db2jcc_javax.jar</code></td>
</tr>
<tr>
<td>Oracle</td>
<td><code>ojdbc14.jar</code></td>
<td></td>
</tr>
<tr>
<td>SQL Server JDBC driver provided by Microsoft</td>
<td><code>sqljdbc.jar</code></td>
<td></td>
</tr>
<tr>
<td>SQL Server JDBC driver provided by DataDirect</td>
<td><code>sqlserver.jar;base.jar;util.jar</code></td>
<td></td>
</tr>
</tbody>
</table>
Perform the following steps: add the library paths to the **VMM_JDBC_CLASSPATH** variable:

A. Logon to the WebSphere Application Server Administrative Console.

B. Click **Environment > WebSphere Variables**.

C. Select **scope: cell**.

D. Select the **VMM_JDBC_CLASSPATH** variable or click **New** to create the variable if it does not exist.

E. Enter the complete paths to the library files, separated by ';', in the **Value** field; for example, enter

```plaintext
D:\IBM\SQLLIB\java\db2jcc.jar;D:\IBM\SQLLIB\java\db2jcc_license_cu.jar.
```

Copy the above library files into the **appserver/lib** directory. Then stop and restart the server1 and WebSphere_Portal servers to load the library files. In a clustered environment, you must also stop and restart the Deployment Manager and the nodeagents.

3. Use a text editor to open the **wkplc.properties** file, located in the **wp_profile_root\ConfigEngine\properties** directory.

4. Enter a value for the following required parameters in the **wkplc.properties** file under the VMM Federated Database Properties heading: **Note**: See the properties file for specific information about the required parameters and for advanced parameters.

```plaintext
- federated.db.DataSourceName
- federated.db.DbType
- federated.db.DbUrl
- federated.db.id
- federated.db.baseDN
- federated.db.DbUser
- federated.db.DbPassword
- federated.db.DbName
```

5. Change the value for the **com.ibm.SOAP.requestTimeout** parameter to 1000.

A. Navigate to the following directory: **wp_profile_root\properties**

B. Locate and open **soap.client.props** with any text editor.

C. Locate the **com.ibm.SOAP.requestTimeout** parameter and change the value to 1000.

D. Save and close **soap.client.props**.

6. Perform the following steps in a clustered environment:

A. Run the **ConfigEngine.bat** **wp-prep-vmm-db-secured-environment -DWasPassword=password -** **DbDomain=federated.db -Db_type.DmgrDbLibrary=local full path of the database jars on the Deployment Manager -DmgrNodeName=dmgr_node_name** task from the **wp_profile_root\ConfigEngine** directory to create the local Deployment Manager WebSphere variable used to access the database jars. **Note**: The **db_type** in **db_type .DmgrDbLibrary** should be set to the type of database you are using, for example **db2**. The **local full path of the database jars on the Deployment Manager** should be one of the following options:

- **DB2 Type 2 driver**: db2java.zip
- **DB2 Type 4 driver**: db2jcc.jar;db2jcc_license_cu.jar
- **DB2 for z/OS Type 2 driver**: db2java.zip
- **DB2 for z/OS Type 4 driver**: db2jcc.jar;db2jcc_license_cisuz.jar;db2jcc_javax.jar
- **Oracle**: ojdbc14.jar
- **SQL Server JDBC driver provided by Microsoft**: sqljdbc.jar
SQL Server JDBC driver provided by DataDirect: sqlserver.jar; base.jar; util.jar

B. Run the following task. Include each node name as a comma separated list in the command:
1. Set the property value for `federated.db.DbType` in `wkplc.properties` if you are using a database user registry.
2. Run the `ConfigEngine.bat wp-node-prep-vmm-db-secured-environment` task from the `wp_profile_root\ConfigEngine` directory on each node to create the variable used to access the VMM database jars. **Note:** `VmmNodeName` is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The `db_type` in `db_type.NodeDbLibrary` should be set to the type of database you are using, for example `db2`.

C. Stop and restart all necessary servers to propagate your changes.

7. Run the `ConfigEngine.bat wp-create-db` task from the `wp_profile_root\ConfigEngine` directory, to add a database user registry to the default federated repository. **Note:** Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

9. Perform the following steps to update the user registry where new users and groups are stored:
   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root\ConfigEngine\properties` directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - `personAccountParent`
      - `groupParent`
      - `personAccountRdnProperties`
      - `groupRdnProperties`

      The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:

      - `personAccountParent=dc=yourco,dc=com`
      - `groupParent=dc=yourco,dc=com`

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `ConfigEngine.bat wp-set-entitytypes` task, from the `wp_profile_root\ConfigEngine` directory, to delete the old attributes before adding the new attributes.

   E. Stop and restart all necessary servers to propagate your changes.

10. **Optional:** Run the `ConfigEngine.bat wp-query-repository` task, from the `wp_profile_root\ConfigEngine` directory, to list the names and types of configured repositories.

**Parent topic:** Configuring the default federated repository on Windows

**Parent topic:** Updating your user registry on Windows
Adding realm support on Windows

A realm is a group of users from one or more user registries that form a coherent group within IBM® WebSphere® Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

Before configuring realm support, you must add all LDAP user registries and/or database user registries, that you will use to create a single realm or multiple realms, to the federated repository. If you are going to create multiple realms, you must create all required base entries within your LDAP user registries and/or database user registries. All base entry names must be unique within the federated repository. In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add realm support to your user registry model:

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root\ConfigEngine\properties directory.
2. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading. **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - realmName
   - securityUse
   - delimiter
   - addBaseEntry
3. Save your changes to the wkplc.properties file.
4. Run the ConfigEngine.bat wp-create-realm -DWasPassword=password task, from the wp_profile_root\ConfigEngine directory, to add a new realm to the Virtual Member Manager configuration. **Important:** To create multiple realms, ensure that your federated repository contains the required unique base entries. Stop and restart the appropriate servers for your installation environment, and then update the wkplc.properties file with the base entry information and rerun the wp-create-realm task. Repeat these steps until all realms are created.
5. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.
6. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading and then save your changes. **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - realmName
   - realm.personAccountParent
   - realm.groupParent
   - realm.orgContainerParent
7. Run the ConfigEngine.bat wp-modify-realm-defaultparents -DWasPassword=password task, from the wp_profile_root\ConfigEngine directory, to update the default parents per entity type and realm. **Important:** Stop and restart the appropriate servers for your installation environment before rerunning this task for any additional entity types and realms.
8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

9. Optional: Perform the following steps to add additional base entries to the realm configuration; for example, if you had two additional base entries (base entry 1 and base entry 2) to add to the realm you just created, you would update the wkplc.properties file with the information from base entry 1 and then run this task. Then you would update the properties file with the information for base entry 2 and then run this task:
   A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root\ConfigEngine\properties directory.
   B. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading:
      - realmName
      - addBaseEntry
   C. Save your changes to the wkplc.properties file.
   D. Run the ConfigEngine.bat wp-add-realm-baseentry -DWasPassword=password task, from the wp_profile_root\ConfigEngine directory, to add an additional LDAP base entries to the realm configuration.
   E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID; this step is required if you change the default realm:
   A. Create a new user in the Manage Users and Groups portlet to replace the current WebSphere Application Server administrative user.
   B. Create a new user in the Manage Users and Groups portlet to replace the current WebSphere Portal administrative user.
   C. Create a new group in the Manage Users and Groups portlet to replace the current group.
   D. Run the ConfigEngine.bat wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid task, from the wp_profile_root\ConfigEngine directory, to replace the old WebSphere Application Server administrative user ID and group ID with the new user and group. **Important:** You must provide the full distinguished name (DN) for the newAdminId and newAdminGroupId parameters.
   E. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

11. Optional: Perform the following steps to set the realm you created as the default realm: **Remember:** Only users defined in base entries that exist in the default realm are able to log into WebSphere Portal. If you find that a user cannot log in to WebSphere Portal, check to see if the base entry that contains the user exists in the default realm. You can run the wp-query-realm-baseentry task to see what base entries are part of the default realm. If the default realm is missing the base entry, run the wp-add-realm-baseentry task to add the base entry to the default realm.
A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root\ConfigEngine\properties directory.
B. For defaultRealmName, type the realmName property value you want to use as the default realm.
C. Save your changes to the wkplc.properties file.
D. Run the ConfigEngine.bat wp-default-realm -DWasPassword=password task, from the wp_profile_root\ConfigEngine directory, to set this realm as the default realm.
E. Stop and restart all necessary servers to propagate your changes.

12. Optional: Perform the following steps to query a realm for a list of its base entries:
A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root\ConfigEngine\properties directory.
B. For realmName, type the name of the realm you want to query.
C. Save your changes to the wkplc.properties file.
D. Run the ConfigEngine.bat wp-query-realm-baseentry -DWasPassword=password task, from the wp_profile_root\ConfigEngine directory, to list the base entries for a specific realm.

13. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:
A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root\ConfigEngine\properties directory.
B. Enter a value for realmName or leave blank to update the default realm.
C. Save your changes to the wkplc.properties file.
D. Run the ConfigEngine.bat wp-modify-realm-enable-dn-login -DWasPassword=password task, located in the wp_profile_root\ConfigEngine directory, to enable the distinguished name login. Note: After running this task to enable the full distinguished name login, you can run the ConfigEngine.bat wp-modify-realm-disable-dn-login -DWasPassword=password task to disable the feature.
E. Stop and restart all necessary servers to propagate your changes.

Parent topic: Configuring the default federated repository on Windows
Adapting the attribute configuration on Windows

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

After installation, IBM WebSphere Portal has a predefined set of attributes for users and groups. Your LDAP server may have a different set of predefined user and group attributes. To ensure proper communication between WebSphere Portal and your LDAP server, you can configure additional attributes and flag existing attributes as required or unsupported on a per repository basis or for all configure repositories.

LDAP servers can only handle attributes that are explicitly defined in their schema. The LDAP server’s schema is different from the WebSphere Portal schema but the two schemas should match for proper communication between WebSphere Portal and the LDAP server. The task to add the LDAP user registry does some basic attribute configurations depending on the type of LDAP server that you choose. You may, however, still need to adapt the WebSphere Portal configuration to match the LDAP schema; for example, if an attribute is defined in WebSphere Portal but not in the LDAP server, you will need to perform one of the following tasks to resolve this mismatch:

- Flag the attribute as unsupported for the LDAP server
- Introduce an attribute mapping that maps the WebSphere Portal attribute to an attribute defined in the LDAP schema

Perform the following tasks to adapt the attribute configuration to match the configured LDAP server(s) and your business needs:

1. Querying the defined attributes on Windows
   After installing IBM WebSphere Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

2. Adding attributes on Windows
   The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM WebSphere Portal and your user registry.

3. Mapping attributes
   After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

4. Removing attributes
   Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

Parent topic: Configuring WebSphere Portal to use a user registry on Windows
Previous topic: Choosing your user registry model on Windows
Next topic: Stand-alone server: Configuring WebSphere Portal to use dynamic groups on Windows

Related tasks
Adding an LDAP user registry on Windows
Adding an LDAP user registry over SSL on Windows
Configuring a stand-alone LDAP user registry on Windows
Configuring a stand-alone LDAP user registry over SSL on Windows
Querying the defined attributes on Windows

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

**Warning:** If you are using a database user registry or a property extension database, copy the database drivers to the `AppServer_root/lib` directory before executing this script.

Run the `ConfigEngine.bat wp-query-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, any time during the configuration process or at runtime to query an overview of the currently defined attributes. This task creates the `availableAttributes.html` report, located in the `wp_profile_root/ConfigEngine/log` directory. The report contains one table that lists the available attributes for Users (**PersonAccount**) and one table that lists the available attributes for Groups. For each configured repository there is a column that indicates if the attribute is flagged as unsupported or if the attribute is mapped to a different LDAP attribute. **Note:** This task does not validate the existence of attributes in the LDAP schema.

**Parent topic:** Adapting the attribute configuration

**Next topic:** Adding attributes on Windows
Adding attributes on Windows

The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM® WebSphere® Portal and your user registry.

Perform the following steps to add new attributes to your user registry:

1. Install the required Enterprise Archive (.ear) file on WebSphere Application Server. Select the appropriate task to install the .ear file:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone environment</td>
<td>Perform the following steps to install the .ear file in a stand-alone environment: Open a command prompt. Navigate to the wp_profile_root\ConfigEngine directory. Run the ConfigEngine.bat wp-la-install-ear -DWasPassword=password task.</td>
</tr>
<tr>
<td>Clustered environment</td>
<td>Perform the following steps to install the .ear file in a clustered environment: Open a command prompt. Navigate to the wp_profile_root\ConfigEngine directory on the primary node. Run the ConfigEngine.bat wp-la-install-ear -DWasPassword=dmgr_password -DServerName=dmgr -DNodeName=node_name task. Where node_name is the name of the node where the deployment manager resides; you can find the node_name value in the WebSphere Application Server Administrative Console under System administrator &gt; Deployment Manager &gt; Runtime tab &gt; General Properties &gt; Node Name.</td>
</tr>
</tbody>
</table>

2. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: [Starting and stopping servers, deployment managers, and node agents](#).

3. Use a text editor to open the wkplc.properties file, located in the wp_profile_root\ConfigEngine\properties directory.

4. Enter a value for the following required parameters in the wkplc.properties file under the VMM Property Extension Properties heading:

   * **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - la.providerURL
   - la.propertyName
   - la.entityTypes
   - la.dataType
   - la.multiValued

5. Save your changes to the wkplc.properties file.

6. Run the ConfigEngine.bat wp-add-property -DWasPassword=password task to add the attribute to the user registry.

   **Note:** This task performs an EJB call to WebSphere Application Server, which must authenticate against WebSphere.
Application Server. Depending on the configuration in the `sas.client.props` file, you may receive a popup window or a command line prompt asking for user identity and password. Enter the WebSphere Application Server user ID and password.

**Remember:** If you have multiple properties to add, repeat all steps, except for the `wp-la-install-ear` task, until all new attributes are added.

7. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents.*

**Parent topic:** Adapting the attribute configuration  
**Previous topic:** Querying the defined attributes on Windows  
**Next topic:** Mapping attributes

**Related tasks**
Starting and stopping servers, deployment managers, and node agents
Mapping attributes

After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

Perform the following steps to map attributes between WebSphere Portal and your LDAP server; if you have multiple LDAP servers, you will need to perform these steps for each LDAP server:

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root\ConfigEngine\properties directory.

2. Enter a value for one of the following sets of parameters in the wkplc.properties file to identify your LDAP server:
   
   **Note:** Make sure you use the same values you used to configure your LDAP server.

   Table 1. Identifying your LDAP server in the wkplc.properties file.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>The following parameters are found under the LDAP attribute configuration heading. <strong>Note:</strong> See the properties file for specific information about the required parameters and for advanced parameters.</td>
</tr>
<tr>
<td></td>
<td>standalone.ldap.idstandalone.ldap.hoststandalone.ldap.portstandalone.ldap.sslEnabledstandalone.ldap.bindDNstandalone.ldap.bindPasswordstandalone.ldap.baseDN</td>
</tr>
<tr>
<td>Federated</td>
<td>The following parameters are found under the VMM Federated repository properties heading. <strong>Note:</strong> See the properties file for specific information about the required parameters and for advanced parameters.</td>
</tr>
<tr>
<td></td>
<td>federated.ldap.idfederated.ldap.hostfederated.ldap.portfederated.ldap.sslEnabledfederated.ldap.bindDNfederated.ldap.bindPasswordfederated.ldap.baseDN</td>
</tr>
</tbody>
</table>

3. Run one of the following tasks to check that all defined attributes are available in the configured LDAP user registry:

   Table 2. Task to check that all defined attributes are available in the configured LDAP user registry.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>ConfigEngine.bat wp-validate-standalone-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root\ConfigEngine directory.</td>
</tr>
<tr>
<td>Federated</td>
<td>ConfigEngine.bat wp-validate-federated-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root\ConfigEngine directory.</td>
</tr>
</tbody>
</table>

4. Open the ConfigTrace.log file, located in the wp_profile_root\ConfigEngine\log directory, to review the following output for the PersonAccount and Group entity type:
- The following attributes are defined in WebSphere Portal but not in the LDAP server
  - This list contains all attributes that are defined in WebSphere Portal but not available in the LDAP. Flag attributes that you do not plan to use in WebSphere Portal as unsupported. Map the attributes that you plan to use to the attributes that exist in the LDAP; you must also map the uid, cn, firstName, sn, preferredLanguage, and ibm-primaryEmail attributes if they are contained in the list.

- The following attributes are flagged as required in the LDAP server but not in WebSphere Portal
  - This list contains all attributes that are defined as "MUST" in the LDAP server but not as required in WebSphere Portal. You should flag these attributes as required within WebSphere Portal; see the step below about flagging an attribute as either unsupported or required.

- The following attributes have a different type in WebSphere Portal and in the LDAP server
  - This list contains all attributes that WebSphere Portal might ignore because the data type within WebSphere Portal and within the LDAP server do not match.

5. Use a text editor to open the wkplc.properties file, located in the wp_profile_root\ConfigEngine\properties directory.

6. Enter a value for one of the following sets of parameters in the wkplc.properties file to correct any issues found in the config trace file:

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Parameters</th>
</tr>
</thead>
</table>
| Stand-alone     | The following parameters are found under the LDAP attribute configuration heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.  
  standalone.ldap.idstandalone.ldap.attributes.nonSupportedstandalone.ldap.attributes.nonSupported.deletestandalone.ldap.attributes.mapping.ldapNamestandalone.ldap.attributes.mapping.p ortalNamestandalone.ldap.attributes.mapping.entityTypes  
  For example, the following values will flag certificate and members as unsupported attributes and will map ibm-primaryEmail to mail and ibm-jobTitle to title for both the PersonAccount and Group entityTypes:  
  standalone.ldap.attributes.nonSupported=certificate, members  
  standalone.ldap.attributes.nonSupported.delete=standalone.ldap.attributes.mapping.ldapName=mail, title  
  standalone.ldap.attributes.mapping.p ortalName=ibm-primaryEmail, ibm-jobTitle  
  standalone.ldap.attributes.mapping.entityTypes=PersonAccount, Group |
7. Save your changes to the wkplc.properties file.

8. Run one of the following tasks to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between WebSphere Portal and the LDAP user registry:

Table 4. Task to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between Portal and the LDAP user registry.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>ConfigEngine.bat wp-update-standalone-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root\ConfigEngine directory</td>
</tr>
<tr>
<td>Federated</td>
<td>ConfigEngine.bat wp-update-federated-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root\ConfigEngine directory</td>
</tr>
</tbody>
</table>

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

10. Optional: Perform the following steps to flag an attribute as either unsupported or required for the entire WebSphere Portal environment instead of just for the specified LDAP:

A. Enter a value for the following required parameters in the wkplc.properties file:

```
user.attributes.required
user.attributes.nonsupported
```

B. Save your changes to the wkplc.properties file.

C. Run the ConfigEngine.bat wp-update-attribute-config -DWasPassword=password task, from the wp_profile_root\ConfigEngine directory.

D. Stop and restart all necessary servers to propagate your changes.

**Parent topic:** Adapting the attribute configuration
Removing attributes

Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

Perform the following steps to remove an attribute from your database: **Important:** Do not remove attributes that have already been populated with user values because this can cause database inconsistencies.

**Cluster note:** In a clustered environment, perform the following steps on the deployment manager and then resynch the nodes.

1. Perform the following steps to remove an attribute stored in a property extension database; this is an attribute that was added using the `wp-add-la-property` task.
   
   A. Open the tool you use to edit your database.
   
   B. Verify that your attribute name is available in the LAPROP table.
   
   C. Delete the required attributes from the LAPROP table.
   
   D. Open the `wimxmlextension.xml` file, located in the `wp_profile_root/config/cells/cellname/wim/model` directory.
   
   E. Locate and delete the `propertySchema` definition for the attributes that you deleted from the LAPROP table; for example:
   
   ```xml
   <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String"
   multiValued="true" propertyName="attribute_name">
   <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
   </wim:propertySchema>
   ```
   
   F. Save your changes to the `wimxmlextension.xml` file.
   
   G. Open the `wimconfig.xml` file, located in the `wp_profile_root/config/cells/cellname/wim/config` directory.
   
   H. Locate and delete the `propertiesNotSupported` definitions for the attributes that you deleted from the LAPROP table; for example:
   
   ```xml
   <config:propertiesNotSupported name="attribute_name">
   </config:propertiesNotSupported>
   ```
   
   I. Save your changes to the `wimconfig.xml` file.
   
   J. Stop and restart the server1 and WebSphere_Portal servers from the `wp_profile_root/bin` directory.

2. Perform the following steps to remove an attribute that is not stored in a property extension database:
   
   A. Open the `wimxmlextension.xml` file.
   
   B. Locate and delete the `propertySchema` definition for the attributes you previously added; for example:
   
   ```xml
   <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String"
   multiValued="true" propertyName="attribute_name">
   <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
   </wim:propertySchema>
   ```
   
   C. Save your changes to the `wimxmlextension.xml` file.
   
   D. Open the `wimconfig.xml` file.
   
   E. Locate and delete the stanza that corresponds to the custom attribute you deleted from the `wimextension.xml` file; for example:
   
   ```xml
   <config:attributes name="attribute_name" propertyName="property_name">
   <config:entityTypes>PersonAccount</config:entityTypes>
   </config:attributes>
   ```
   
   F. Save your changes to the `wimconfig.xml` file.
G. Stop and restart the server1 and WebSphere_Portal servers.

**Parent topic:** Adapting the attribute configuration  
**Previous topic:** Mapping attributes
Stand-alone server: Configuring WebSphere Portal to use dynamic groups on Windows

By default, WebSphere Portal is enabled for static groups. However, the Virtual Member Manager (VMM) allows users to be members of either static or dynamic groups. Static groups are those where a persistent binding exists between a group and its members. Dynamic groups are those where a search query is defined to retrieve the members of a group. If you have your LDAP server configured to use dynamic groups, complete the steps in this task for WebSphere Portal to use dynamic group queries when you setup your LDAP server.

Perform the required tasks to configure either a stand-alone or federated LDAP server security.

The steps in this task use groupOfURLs as the object class for dynamic groups and memberURL as the dynamic membership attribute. The actual values for object classes and dynamic membership attributes can vary depending on your LDAP server. For this reason, you should export an LDIF file to verify the object classes and dynamic membership attributes. Either refer to your LDAP documentation or ask your LDAP administrator for instructions on exporting an LDIF file.

Clustered environments: Perform the following steps on the Deployment Manager then synchronize the nodes.

To configure WebSphere Portal to use dynamic groups, do the following:

1. Choose the appropriate set of steps, depending on your LDAP server environment: Table 1. Steps for enabling dynamic groups

<table>
<thead>
<tr>
<th>LDAP server environment</th>
<th>Steps to perform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone LDAP server or federated LDAP server(s)</td>
<td>Navigate to the following directory: wp_profile_root/cells/cell_name/wim/config. Locate and open wimconfig.xml with any text editor. Add the following line to the <a href="">config:groupConfiguration</a> tag: <code>&lt;config:dynamicMemberAttributes name=&quot;memberurl&quot; objectClass=&quot;groupofurls&quot;/&gt;</code> Save and close wimconfig.xml.</td>
</tr>
<tr>
<td>Federated LDAP server(s)</td>
<td>Log in to the administration console. Select Security &gt; Secure administration, applications, and infrastructure. Under Available realm definitions, select Federated repositories and click Configure. Under Related Items, click Manage repositories. Select the appropriate repository from the list. Under Additional Properties, click Group attribute definition then click Dynamic member attributes. Click New and specify values for the Name and Object class fields as appropriate. For example, Name: memberurl Object class: groupofurls Click OK and save the changes to the master configuration.</td>
</tr>
</tbody>
</table>

2. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.
Stand-alone server: Enabling referrals for your LDAP user registry on Windows

Referrals redirect object requests from one LDAP server to another when objects do not exist or cannot be located in a particular directory tree. You should enable referrals if your environment has more than one user registry existing on multiple servers or domains.

To configure your portal to use LDAP referrals, do the following:

1. Use any text editor to open the `wkplc.properties` file in the following directory: `wp_profile_root/ConfigEngine/properties`.

2. Specify values for the following parameters:
   - `et.ldap.id=ID_of_your_LDAP_server`
   - `et.ldap.host=hostname_of_your_LDAP_server`
   - `et.ldap.referral=follow`

3. Save and close `wkplc.properties`.

4. Run the following task from the `wp_profile_root/ConfigEngine` directory to create an LDAP entity type:
   - UNIX: `./ConfigEngine.sh wp-update-et-ldap -DWasPassword=password`
   - Windows: `ConfigEngine.bat wp-update-et-ldap -DWasPassword=password`
   - IBM® i5/OS: `ConfigEngine.sh wp-update-et-ldap -DWasPassword=password`

5. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

Parent topic: Configuring WebSphere Portal to use a user registry on Windows

Previous topic: Stand-alone server: Configuring WebSphere Portal to use dynamic groups on Windows
Tune your servers

Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

- Prerequisites
  - Setting up a database
  - Base Portal Tuning Scenarios

Tuning a WebSphere Portal environment involves tuning and configuring the various systems and components of the environment. The tuning guide provides general concepts and details specifics configuration instructions. Instructions are included for:

- Configuring the application server and the resources defined for that application server
- Determining the cloning strategy for expanding or extending the environment
- Tuning the database(s) and database server
- Tuning the directory server and its database
- Tuning the Web server
- Tuning the operating system and network
- Tuning the WebSphere Portal services

Parent topic: Setting up a stand-alone server on Windows
Previous topic: Configuring WebSphere Portal to use a user registry on Windows

Related Information
- WebSphere Portal and Lotus Web Content Management 6.1.x Performance Tuning Guide
- IBM WebSphere Portal Performance Troubleshooting Guide
Setting up a cluster

Clusters enable you to scale your WebSphere Portal configuration. Clusters also enable enterprise applications to be highly available because requests are automatically routed to the running servers in the event of a failure. There are numerous cluster configuration, such as horizontal, vertical, multiple, and dynamic.

**Prerequisites**

- Technotes for installation and configuration issues

The following illustration depicts a horizontal cluster configuration where IBM® WebSphere® Portal is installed on multiple servers. This configuration reduces single points of failure, but requires more hardware.

To conserve hardware, you can also set up a vertical cluster with multiple portal instances on a single node.
In response to customer requests, the instructions to setup WebSphere Portal has been improved to provide operating system specific instructions for setting up a production environment. Select your operating system to get started.

After completing the installation remember to tune the servers in your portal environment in order to achieve better performance. The Base Portal Tuning scenarios in the WebSphere Portal and Lotus Web Content Management 6.1.x Performance Tuning Guide provides information about tuning the WebSphere Application Server, WebSphere Portal services, databases, directory servers and more. Base Portal Tuning scenarios are the foundation to most performance tuning. In a cluster environment, there are additional steps you should take to tune WebSphere Application Server, the Web server, and more. First review and take the necessary steps provided in the Base Portal Tuning scenarios, then review and take additional necessary steps in the Tuning a Cluster Environment chapter of the tuning guide. Base Portal Tuning scenarios

Tuning a cluster environment

- **Setting up a cluster on AIX**
  In a production environment, you can install and configure IBM WebSphere Portal on a single (primary) node and then add additional nodes in the future as your company's needs grow. You can also deploy a high availability production environment by creating and configuring clustered deployments of multiple servers running WebSphere Portal managed by IBM WebSphere Application Server Network Deployment.

- **Setting up a cluster on HP-UX**
  In a production environment, you can install and configure IBM WebSphere Portal on a single (primary) node and then add additional nodes in the future as your company's needs grow. You can also deploy a high availability production environment by creating and configuring clustered deployments of multiple servers running WebSphere Portal managed by IBM WebSphere Application Server Network Deployment.

- **Setting up a cluster on IBM i5/OS**
  In a production environment, you can install and configure IBM WebSphere Portal on a single (primary) node and then add additional nodes in the future as your company's needs grow. You can also deploy a high availability production environment by creating and configuring clustered deployments of multiple servers running WebSphere Portal managed by IBM WebSphere Application Server Network Deployment.
- **Setting up a cluster on Linux**
  In a production environment, you can install and configure IBM WebSphere Portal on a single (primary) node and then add additional nodes in the future as your company's needs grow. You can also deploy a high availability production environment by creating and configuring clustered deployments of multiple servers running WebSphere Portal managed by IBM WebSphere Application Server Network Deployment.

- **Setting up a cluster on Solaris**
  In a production environment, you can install and configure IBM WebSphere Portal on a single (primary) node and then add additional nodes in the future as your company's needs grow. You can also deploy a high availability production environment by creating and configuring clustered deployments of multiple servers running WebSphere Portal managed by IBM WebSphere Application Server Network Deployment.

- **Setting up a cluster on Windows**
  In a production environment, you can install and configure IBM WebSphere Portal on a single (primary) node and then add additional nodes in the future as your company's needs grow. You can also deploy a high availability production environment by creating and configuring clustered deployments of multiple servers running WebSphere Portal managed by IBM WebSphere Application Server Network Deployment.

**Parent topic:** Installing WebSphere Portal

**Related concepts**

Cluster considerations
Understanding single-cell deployment scenarios
Web content authoring environments

**Related tasks**

Adding a BPI-enabled portal server to a managed cell in a single-cell setup

**Related Information**

*Secondary node fails to start after being added to the cluster with javax.jcr.RepositoryException exception*
Setting up a cluster on AIX

In a production environment, you can install and configure IBM® WebSphere® Portal on a single (primary) node and then add additional nodes in the future as your company’s needs grow. You can also deploy a high availability production environment by creating and configuring clustered deployments of multiple servers running WebSphere Portal managed by IBM WebSphere Application Server Network Deployment.

- **Prerequisites**
  - [Technotes for installation and configuration issues](#)

Perform the following tasks to set up your production environment on AIX:

1. **Preparing prerequisite and corequisite software on AIX**
   Before creating your high availability production environment, you must prepare your database, your user registry, your deployment manager, and your remote Web server.

2. **Preparing your AIX operating system**
   View information on setting up your operating system for IBM WebSphere Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

3. **Preparing the primary node on AIX**
   Before creating your high availability environment, you must install IBM WebSphere Portal on your primary node and then configure your database and your network deployment manager.

4. **Choosing the type of cluster to create on AIX**
   If you installed a IBM WebSphere Application Server Network Deployment, create a static cluster to handle failover requests. If you installed a IBM WebSphere Virtual Enterprise, create a dynamic cluster to balance member workloads.

5. **Preparing a remote Web server when portal is installed on AIX**
   Install and configure the Web server plug-in, provided by IBM WebSphere Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

6. **Configuring WebSphere Portal to use a user registry on AIX in a clustered environment**
   Configure user registry security on IBM WebSphere Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

7. **Preparing additional nodes on AIX**
   After installing and configuring your primary node, you can create your secondary nodes. You must install IBM WebSphere Portal on each node and then configure the node to access the database and user registry before adding it to the cluster.

8. **Tune your servers**
   Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

9. **Configuring search in a cluster on AIX**
   IBM WebSphere Portal provides two distinct search capabilities. You can utilize both types of search capabilities in a clustered environment.
10. **Setting up multiple clusters on AIX**

The majority of the steps to build another cluster are the same as when building the first cluster in the same cell, with a few exceptions. Basically, the new profile will be designated as the primary profile, using IBM WebSphere Portal clustering terminology, and will then be used as the basis for the new cluster definition. This duplicates the build process of the first cluster in the cell. During the federation process, if any applications on this new node already exist in the cell (because they are in use by the first cluster), then Deployment Manager will not allow them to be added. After federation, the applications that already exist in the cell are not mapped to the WebSphere_Portal server on the newly federated node, and thus the existing applications must be remapped to this newly federated server to restore its application list. Therefore, depending on the configuration of the new profile, there will likely be some combination of applications shared with the other existing cluster, and some applications unique to this new profile.

11. **Sharing database domains between clusters on AIX**

To help provide redundancy and failover support in production environments composed of multiple clusters in a single cell and multiple clusters in different cells, you can share database domains between those clusters. IBM WebSphere Portal data is organized into different database domains, with different availability requirements depending on how the production environment is set up. When multiple lines of production are involved and each line of production is implemented as a cluster of servers, sharing database domains ensures that the data is automatically synchronized across the lines of production.

**Parent topic:** Setting up a cluster
Preparing prerequisite and corequisite software on AIX

Before creating your high availability production environment, you must prepare your database, your user registry, your deployment manager, and your remote Web server.

Perform the following tasks to prepare prerequisite and corequisite software:

1. **Preparing the WebSphere Application Server Deployment Manager on AIX**
   IBM WebSphere Portal provides a customized installation package (CIP) that includes IBM WebSphere Application Server Network Deployment and all required maintenance packages. Use the supplied discs to install WebSphere Application Server Network Deployment on a dedicated system.

2. **Preparing a remote Web server when portal is installed on AIX**
   Install and configure the Web server plug-in, provided by IBM WebSphere Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

Parent topic: Setting up a cluster on AIX
Next topic: Preparing your AIX operating system
Preparing the WebSphere Application Server Deployment Manager on AIX

IBM® WebSphere® Portal provides a customized installation package (CIP) that includes IBM WebSphere Application Server Network Deployment and all required maintenance packages. Use the supplied discs to install WebSphere Application Server Network Deployment on a dedicated system.

Perform the following steps to prepare the WebSphere Application Server Deployment Manager:

1. Install WebSphere Application Server Network Deployment or use the CIP to upgrade an existing installation. Run the following command: `cd_root/AIX/architecture/ifpackage/WAS/install`, where `cd_root` is the root directory of the disc and `architecture` is the system’s processor architecture.

2. Choose one of the following options to create a default deployment manager profile:
   **Important:** While creating the default deployment manager profile, enable administrative security. If you use the Profile Management Tool, check the enable administrative security check box. If you use the `manageprofile` command, add the `-enableAdminSecurity=true` parameter to the command line.

3. Run the following command to start the deployment manager: `./startManager.sh`, from the `dmgr_profile_root/bin` directory.

4. Use the following URL to launch the network deployment administrative console: `http://dmgr_hostname:9060/ibm/console`, where `dmgr_hostname` is the fully qualified host name for the WebSphere Application Server Network Deployment.

5. Log into the deployment manager administrative console.

6. Increase the HTTP connection timeouts for the deployment manager.
   A. Click System Administration > Deployment Manager > Web container transport chains.
   B. Increase the timeout values. For the `WCInboundAdmin` and `WCInboundAdminSecure` entries listed in the web container transport chains section, complete the following steps to increase the timeout values:
      1. Click HTTP Inbound Channel.
      2. Change the Read timeout value to 180.
      3. Change the Write timeout value to 180.
      4. Save the configuration changes.

7. Change the timeout request period for the Java Management Extensions (JMX) connector.
   A. Click System administration > Deployment Manager > Administration Services > JMX connectors > SOAPConnector > Custom Properties.
   B. Select the requestTimeout property, and increase the value from 600 to 6000.
   C. Save the configuration changes.

8. Update the maximum Java heap size used by the deployment manager:

---

### Table of Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile Management Tool</td>
<td>See Creating a deployment manager profile for information.</td>
</tr>
<tr>
<td>manageprofile commands</td>
<td>See manageprofile commands for information. Note: If you have a 64-bit environment, only the manageprofiles command is supported when creating profiles.</td>
</tr>
</tbody>
</table>

---

**Operating systems:** AIX, HP-UX, i5/OS, Linux, Solaris, Windows
B. Update the value in the **Maximum Heap Size** field. For information about appropriate heap sizes see the documentation for your operating system in the Performance Guides located on the WebSphere Portal and Web Content Management Product Documentation page. **Note:** If using a 32-bit operating system, you will need to set the heap size to a lower size than a 64-bit operating system.

C. Click **OK** and then save your changes.

9. Depending on your version of WebSphere Application Server, make the appropriate selection from the navigation:

   **Note:** If security is not enabled on your deployment manager, see Enabling security for information before performing this step.

   - For **WebSphere Application Server Version 6.1**: Click **Security > Secure administration, applications and infrastructure** and select **Enable Application Security**. Then save the configuration changes.
   
   - For **WebSphere Application Server Version 7.0**: Click **Security > Global security** and select **Enable Application Security**. Then save the configuration changes.

10. Verify that the WebSphere Portal administrative users and administrative group exist in the Deployment Manager cell's user registry. Perform the following steps if you need to create the administrative users and group:

    A. Click **Users and Groups > Manage Users**.
    
    B. Click **Create**.
    
    C. Type the information for the WebSphere Portal administrative users; for example wpsadmin and wpsbind, and then click **Create**.
    
    D. Click **Users and Groups > Manage Groups**.
    
    E. Click **Create**.
    
    F. Type wpsadmins as the name of the WebSphere Portal administrative group and then click **Create**.
    
    G. Click the group you just created; for example wpsadmins.
    
    H. Click the **Members** tab.
    
    I. Click **Add Users**.
    
    J. Search for the users.
    
    K. Select the users you want to add to the group.
    
    L. Click **Add** to add the users to the group.
    
    M. Click **Close** when you are done adding users to the group.
    
    N. Log out of the administrative console.


12. Run the following tasks to stop and restart the deployment manager:

    A. **./stopManager.sh -username admin_userid -password admin_password**, from the dmgr_profile_root\bin directory
    
    B. **./startManager.sh**, from the dmgr_profile_root\bin directory

**Parent topic:** Preparing prerequisite and corequisite software on AIX

**Next topic:** Preparing a remote Web server when portal is installed on AIX
Preparing a remote Web server when portal is installed on AIX

Install and configure the Web server plug-in, provided by IBM® WebSphere® Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

- Prerequisites
  - Preparing the WebSphere Application Server Deployment Manager on AIX
  - Preparing prerequisite and corequisite software on AIX
  - Preparing your AIX operating system
  - Preparing the primary node on AIX
  - Choosing the type of cluster to create on AIX

Perform the following steps to install and configure your Web server:

1. Install and configure the Web server; refer to the Web server documentation for information.
2. If using IBM Lotus® Domino®, edit the NOTES.INI file on the Web server. Set the HTTPEnableConnectorHeaders parameter to 1.
3. If using IBM HTTP Server or Apache Server, edit the httpd.conf file on the Web server. Set the AllowEncodedSlashes directive to on; the directive should be added at the root level as a global directive.

4. Stop the Web server.

5. Install and configure the Web server plug-in on the system where the Web server is located using the Plug-ins installation wizard provided with WebSphere Application Server. Refer to Selecting a Web server topology diagram and roadmap for information. **Note:** The WebSphere Application Server Web server instructions are for the default profile. If not using the default profile, refer to Configuring a Web server for a non-default profile.

   **If using WebDAV:** After successfully installing the Web server plug-in, locate and open your plugin-cfg.xml file and set AcceptAllContent to true.

   **Important:** Depending on how you use the Web server, you may need to adjust the ServerIOTimeout value, which defines how long the plug-in should wait for a response from the application. The recommended minimum value is 60 but you may need to adjust this value higher if you are retrieving data from a database. To update this value, locate and open your plugin-cfg.xml file and set ServerIOTimeout to a value that is appropriate for your business needs. For additional information, see Common questions about the Web server plug-in.

6. If you are using Sun Web Server Version 7 update 8, read Technote 1448262 and perform the steps to resolve the HTTP 408/409 error.

7. Start the Web server.

Parent topic: Preparing prerequisite and corequisite software on AIX

Previous topic: Preparing the WebSphere Application Server Deployment Manager on AIX

Parent topic: Setting up a cluster on AIX

Previous topic: Choosing the type of cluster to create on AIX
Preparing your AIX operating system

View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on AIX

Perform the following steps to prepare your AIX machine:

1. Set the file descriptor limit to 10240; for example, `ulimit -n 10240`.

2. **Web Content Management only**: Use the `ulimit -f` command to set the maximum size of files that can be created to be at least the size of the largest file you would need to upload to the content server. The command `ulimit -f unlimited` removes any limit on file size.

3. Install and configure X server on AIX (for example, X-Windows or GNOME) to use the graphical user interface the installation program provides. **Note**: X server is not required if installing with a response file or in console mode.

**Parent topic:** Setting up a cluster on AIX

**Previous topic:** Preparing prerequisite and corequisite software on AIX

**Next topic:** Preparing the primary node on AIX
Preparing the primary node on AIX

Before creating your high availability environment, you must install IBM® WebSphere® Portal on your primary node and then configure your database and your network deployment manager.

- Prerequisites
  - Preparing prerequisite and corequisite software on AIX
  - Preparing your AIX operating system

Perform the following tasks to prepare your primary node:

1. Installing WebSphere Portal on AIX on the primary node
   IBM WebSphere Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

2. Configure WebSphere Portal to use a remote database
   For high availability, using a remote database is preferred. For improved performance databases can be distributed across multiple database servers. Databases should also be configured for failover. Configuring the database for failover is beyond the scope of this documentation. Refer to the database product documentation for the most authoritative guidance about setting up databases for fail over.

3. Configuring the primary node to communicate with the deployment manager on AIX
   After installing IBM WebSphere Application Server Network Deployment and installing IBM WebSphere Portal on the primary node, you must configure the primary node and the deployment manager to communicate with each other.

4. Removing search collections on AIX
   If you plan to use search in a cluster, you must configure a remote search server; see Configuring Portal Search in a cluster for information. If you created any search collections, you will need to recreate them on the remote search server. If your search collection has data, export the collection before deleting it and then import the data on the remote server.

Parent topic: Setting up a cluster on AIX
Previous topic: Preparing your AIX operating system
Next topic: Choosing the type of cluster to create on AIX
Installing WebSphere Portal on AIX on the primary node

IBM® WebSphere® Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

Review the Supported hardware and software requirements for your operating system and ensure that all hardware and software matches the minimum product level before installing WebSphere Portal. Review Installation methods, options, and sources before installing to determine your installation source location and installation method.

Perform the following steps to install WebSphere Portal and create a custom profile for the node that will be added to the cluster. If you already installed WebSphere Portal, you can skip to the last step to create the custom profile.

1. Type `ping yourserver.yourcompany.com` on a command line to verify that your fully qualified host name is properly configured.
2. Type `ping localhost` on a command line to verify that your network is properly configured.
3. If you are installing with an existing WebSphere Application Server instance, ensure it is installed at the supported level and has the following features installed:
   - **Restriction:** Installing WebSphere Portal Version 6.1.0.1 or higher on an existing WebSphere Application Server Version 7.0 instance where WebSphere Portal Version 7.0 was installed is unsupported.
   - **Important:** Besides ensuring that the existing WebSphere Application Server instance is at the supported level, you will also need to ensure that Apache Derby is installed at the supported level before installing WebSphere Portal. See WebSphere Portal detailed system requirements for supported levels.

   - Application Server
   - Administration
     - Scripted Administration
     - Administrative Console
   - Ant and Deployment Tools
     - Deploy Tool
     - Ant Utilities
4. Optional: Perform the following steps if you want to install WebSphere Portal using a non-root user:
   - **Restriction:** Although it is possible to install WebSphere Application Server as a non-root user, there are limitations to this option that can affect WebSphere Portal functionality, which is why you should install WebSphere Application Server as a root user.
   A. Install the current supported version of WebSphere Application Server as a root user.
   B. Open a command prompt and perform the following steps to create a non-root user and to change ownership:

      1. Use the appropriate system commands to create a new group, to create a new user, and to add the new user to the new group.
      2. Run the following tasks to change the rights of the non-root user:
         ```bash
         chmod -R g+rwx /usr/IBM
         chgrp -R group_name/IBM
         chmod -R g+wr /tmp
         ```
C. Launch the installation program per the next step. During the installation, select the existing WebSphere Application Server using the non-root user and set the installation location to a unique location under the path where non-root permissions were granted.

5. Choose one of the following installation commands based on the installation method you want to use to install WebSphere Portal; see Installation Methods for more information:

 Advance Installation parameters: To customize your installation, you can add various parameters to your installation commands; for example, you can specify your port information. See Advanced installation parameters for information.

 Restriction: When defining your installation path, the ONLY supported characters are ASCII letters [A-Z and a-z], numbers [0-9], slashes [/ or \, depending on your operating system], and the dash [-].

 Table 1. Installation task options

<table>
<thead>
<tr>
<th>Installation type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphical user interface</td>
<td>./install.sh</td>
</tr>
<tr>
<td>Console mode</td>
<td>./install.sh -console from the PortalExpress subdirectory</td>
</tr>
<tr>
<td>Silent install</td>
<td>./install.sh -options &quot;path_to_file/response_filename&quot;, where path_to_file is the full path to the response file and response_filename is the name of the file. A sample install response file (installresponse.txt) and a sample uninstall response file (uninstallresponse.txt) are located in the setup CD root directory. Important: Do not place the response file in a path that contains a space and do not put a space in the file name.</td>
</tr>
</tbody>
</table>

 Note: If the installation program does not detect a WebSphere Application Server instance that you know exists, exit the installation program and pass the WebSphere Application Server instance location using the command line; for example, ./install.sh -W was.undetectedWas="/my/WAS/location".

 Note: If the GUI or console mode installation program fails to detect ports for either WebSphere Application Server or WebSphere Portal, a warning message displays and the installer offers another chance to enter the values. If the silent installation fails to detect ports for either WebSphere Application Server or WebSphere Portal, the installer will exit.

6. Verify your installation was successful; access WebSphere Portal using the http://yourserver:yourport/wps/portal format. Run the ./ConfigEngine.sh list-server-ports -DWasPassword=password task from the wp_profile_root/ConfigEngine directory to generate the wp_profile_root/ConfigEngine/log/wp_PortMatrix.txt file that lists WebSphere Portal ports for your installation.

7. Optional: After the WebSphere Portal installation completes successfully, run the ./ConfigEngine.sh configure-express -DPortalAdminPwd=password -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, if you want to enable the sample IBM Lotus Web Content Management content, such as internet and intranet sample sites.

 Restriction: Run the configure-express task before configuring your database, user registry, context root, security, etc. If you ran any tasks other than the install task, do not run this task.

 Note: See http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_ovw.html for tutorials on how to use the sample content.

 The sample content includes:

 - Creates a group called contentAuthors; members of this group are given privileges to create content in the sample
Internet and intranet sites. Navigate to the **Administration** area and then click **Access > Users and Groups**.

- Creates two new Web Content Management Libraries: "Internet Web Content 6.1.0" and "Intranet Web Content 6.1.0".

Navigate to the **Administration** area and then click **Portal Content > Web Content Libraries**.

- Adds a portlet filter and applies the filter to various portlets in the sample Internet and intranet sites. You can see the definition of the filter in the WebSphere Application Server Administration console and examining the custom resources under the **Environment** area.

- Creates two new theme policies: InternetStyle and IntranetStyle. These styles are applied to sample Internet and intranet sites. Navigate to **Theme Customizer** and then select the style.

- Creates several portlet clones of the Web Content Management rendering portlet. These portlet clones are used on sample Internet and intranet sites.

- Creates two virtual portals with context roots of **wps/portal/intranet** and **wps/portal/internet**. These are the sample Internet and intranet sites. Go to **http://yourserver:yourport/wps/portal/internet** and **http://yourserver:yourport/wps/portal/intranet** to access them.

- Creates several sample credential slots, including "Default slot for E-mail", "Default slot for Feeds", "Default slot for Miscellaneous", "Default slot for Web Clipping", and "Default slot for Web Content Management". Navigate to the **Administration** area and then click **Access > Credential Vault > Manage System Vault Slots**.

- Modifies the portlet palette to include some new portlets. Click the **Portlets** link to see the portlet palette.

Optional: If you ran the **configure-express** task, the owner of the items in the Web content libraries containing the Internet and Intranet Site Template content will be listed as *uid=xyzadmin,o=defaultWIMFileBasedRealm*. To update the owner information for these items to correspond to your portal administrator ID, complete the following steps:

A. Edit the `wp_profile_root/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties` file.

B. Add the following line to the file:

   ```
   uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
   ```

   Replace `portal_admin_DN` with the distinguished name of the portal administrator.

C. Save your changes and close the file.

D. Run the `./ConfigEngine.sh action-express-memberfixer -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory.

9. Optional: Perform the following steps if you need to change the ports for WebSphere Application Server or WebSphere Portal:

   **Note:** The starting port parameter is required for a successful completion of the **modify-ports-by-startport** task.

   Once you specify a start port, this port becomes the base for assigning port values. The code increments this value as each port is assigned, which means that the WebSphere Portal ports will be assigned incrementally starting with the port defined with the **modify-ports-by-startport** task.

   A. Stop the server1 and WebSphere_Portal servers.

   B. Run one of the following commands for each server you need to change:

<table>
<thead>
<tr>
<th>Method</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting port number</td>
<td>./ConfigEngine.sh modify-ports-by-startport -DWasPassword=password -DModifyPortsServer=servername -DStartPort=starting port number</td>
</tr>
</tbody>
</table>
C. Restart the server1 and WebSphere_Portal servers.

**Parent topic:** Preparing the primary node on AIX  
**Next topic:** Configure WebSphere Portal to use a remote database

---

**Related concepts**

Installation methods  
IBM Support Assistant Lite for WebSphere Portal

**Related reference**

Advanced installation parameters

---

**Port file**

Sample port files are available on the Setup disc.

```bash
./ConfigEngine.sh modify-ports-by-portsfile -DWasPassword=password -DModifyPortsServer=servername -DPortsFile=full path to ports file
```

The following is an example of the information within a port file although the port values will be different based on your environment:

```plaintext
BOOTSTRAP_ADDRESS=10031  
SOAP_CONNECTOR_ADDRESS=10033  
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=10032  
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=10025  
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=10036  
WC_adminhost=10027  
WC_defaulthost=10029  
WC_adminhost_secure=10039  
WC_defaulthost_secure=10035  
SIB_ENDPOINT_ADDRESS=10026  
SIB_ENDPOINT_SECURE_ADDRESS=10037  
SIB_MQ_ENDPOINT_ADDRESS=10030  
SIB_MQ_ENDPOINT_SECURE_ADDRESS=10028  
ORB_LISTENER_ADDRESS=10034
```
Configure WebSphere Portal to use a remote database

For high availability, using a remote database is preferred. For improved performance databases can be distributed across multiple database servers. Databases should also be configured for failover. Configuring the database for failover is beyond the scope of this documentation. Refer to the database product documentation for the most authoritative guidance about setting up databases for fail over.

Password considerations when transferring data manually

For security reasons, you should not store passwords in the wkplc.properties and wkplc_comp.properties. Edit each of the properties files prior to running a configuration task, inserting the passwords needed for that task. After the task has run, delete all passwords from each file. For more information, see Deleting passwords from configuration scripts.

Alternatively, you can specify the password on the command line using the following syntax:

- Windows: `ConfigEngine.bat task_name -Dpassword_property_key=password_value`
- UNIX: `.ConfigEngine.sh task_name -Dpassword_property_key=password_value`
- i5/OS: `ConfigEngine.sh task_name -Dpassword_property_key=password_value`

As with other properties, each password property must have the -D prefix and be set equal to (=) a value. If you have multiple properties in a single command, use a space character between each -D property=value setting.

- Prerequisites

  - Installing WebSphere Portal on AIX on the primary node
  - Technotes for database connectivity issues

- Configuring WebSphere Portal to use DB2

View information on installing and setting up DB2 to work with WebSphere Portal.

- Configuring WebSphere Portal to use DB2 for z/OS

View information on installing and setting up DB2 for z/OS to work with WebSphere Portal.

- Configuring WebSphere Portal to use Oracle

View information on installing and setting up Oracle to work with WebSphere Portal.

- Configuring WebSphere Portal to use Oracle RAC

View information on installing and setting up Oracle RAC to work with WebSphere Portal.

- Configuring WebSphere Portal to use SQL Server

View information on installing and setting up SQL Server to work with WebSphere Portal.

- Verifying databases

After you configure IBM WebSphere Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.

Parent topic: Preparing the primary node on AIX
Previous topic: Installing WebSphere Portal on AIX on the primary node
Next topic: Configuring the primary node to communicate with the deployment manager on AIX

Related Information
Troubleshooting WebSphere Portal Version 6.1 databases
Configuring WebSphere Portal to use DB2

View information on installing and setting up DB2 to work with WebSphere Portal.

1. Installing DB2
   View information on installing DB2 for use with WebSphere Portal.

2. Create users
   View the steps to set up users for DB2 to work with WebSphere Portal.

3. Creating remote databases
   A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

4. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

5. Modifying database properties
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

6. Optional: Configuring JCR collation support
   View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

7. Configure WebSphere Portal to use DB2
   View information on how to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

8. Configuring DB2 for large file handling in Web Content Management
   If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

9. Optional: Changing driver types
   WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Parent topic: Configure WebSphere Portal to use a remote database
Installing DB2

View information on installing DB2 for use with WebSphere Portal.

Before you begin:

- Make sure DB2 is set up with updated kernel parameters according to the recommendations in the DB2 Quick Beginnings guide at DB2 Technical Support.

- **(For Type 2 drivers only):** DB2 Connect must be installed on the same system as WebSphere Portal, and have the same name as the server profile name.

- When you install DB2 using the DB2 installation program, it automatically creates a DB2 administrative user with the correct operating system rights.

1. If you are using the JDBC driver in type 2 mode, configure your DB2 Connect client with the following commands. If you are using a remote database, complete this step separately from the WebSphere Portal installation.

   - db2 update dbm cfg using tp_mon_name WAS
   - db2 update dbm cfg using spm_name hostname, where hostname is the host name of WebSphere Portal.

     Because the default for spm_name is the hostname itself, specifying the hostname parameter is optional. If your hostname is more than eight characters, use empty double quotes (" "). For example, db2 update dbm cfg using spm_name " ".

2. Before installing DB2, log in with a user ID that has administrative authority.

3. To install DB2, follow the instructions that are provided with the DB2 documentation to install DB2 and the required fix pack.

Parent topic: Configuring WebSphere Portal to use DB2

Next topic: Create users

Related Information

- DB2 Technical Support
- DB2 Information Center
Create users

View the steps to set up users for DB2 to work with WebSphere Portal.
Before you begin: You should have completed Installing DB2.

- Prerequisites
  - Installing DB2

For all databases, use one user with administrative rights on the operating system and the DB2 installation. This user can be the database administrative user that is created automatically by the DB2 installation program. This user is the database configuration user that will be used for configuration tasks: creating database tables and performing database transfer. If you choose to have WebSphere Portal also create the databases, the database user should be given SYSADM rights. You only need to manually create an administrator ID when you do not want to use an existing DB2 administrator ID.

A common user name is `db2inst1`, but you can assign any user name as long as it has administrative access and follows the limitations listed here. Do not change the user name after creating it.

The limitations on user names are:

- User names can contain one to eight characters.
- Group and instance names can contain one to eight characters.
- Names cannot be any of the following: `users`, `admins`, `guests`, `public`, `local`
- Names cannot begin with: `IBM`, `SQL`, `SYS`
- Names cannot include accented characters.
- Create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Following are the basic steps to create new users and grant privileges. For more information, refer to the DB2 product documentation.

1. From the DB2 Control Center, expand the object tree until you find the DB Users folder.
2. Right-click the DB Users folder and select `Create/Add`. The Create DB User notebook opens.
3. In the User name field, type a user ID.
4. Assign an authority for the user.
5. Click `OK`.

Parent topic: Configuring WebSphere Portal to use DB2
Previous topic: Installing DB2
Next topic: Creating remote databases
Creating remote databases

A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

Before creating these databases, note the following information:

- Use IBM® DB2 Universal Database™ Enterprise Server Edition JDBC driver, JCC, type 4.

- If the DB2 JDBC type 4 driver is used, only the instructions on the remote server are required. You do not need to install the DB2 client software, and you do not need to complete the steps related to the DB2 client.

- The client software, DB2 Connect, must be correctly configured to connect to the remote DB2 server instance, for example, `db2inst1`.

- These instructions assume that a remote DB2 server and DB2 Connect are already installed and running. For information on installing DB2 Connect, see the DB2 product documentation.

- **Prerequisites**
  - Installing DB2
  - Create users

To create a database, you must be a DB2 System Administrator with sufficient database privileges (SYSADM or at a minimum SYSCtrl).

1. Log in to the DB2 server system as a user with sufficient database privileges.
2. Initialize a DB2 command environment by opening a command prompt and typing `su - db2inst1`, where `db2inst1` is the database user. The prompt changes to your operating system shell prompt, for example: `$`. In this mode, you must type `db2` at the beginning of each command and use double quotation marks (""") around the entire command. The following example shows how to use the double quotes; it is not an actual command that you run:

   ```sql
   db2 "CREATE REGULAR TABLESPACE SMS PAGESIZE 4 K MANAGED BY SYSTEM USING ('sms') BUFFERPOOL SMSPOOL"
   ```

   **Note:** If you are using the Command Line Processor (CLP), refer to your DB2 documentation for details. The command prompt is `db2=>`. In this mode, commands can be entered without the `db2` prefix or the double quotation marks. However, the following steps assume you are not using the CLP and are entering commands from your operating system shell prompt, for example: `$`.

3. Run the following commands on the DB2 server system to configure the DB2 database instance:

   **Environment** | **Description**
   --- | ---
   **DB2** | `db2set DB2_RA_TO_RS=YES`
   | `db2set DB2_EVALUNCOMMITTED=YES`
   | `db2set DB2_INLIST_TO_NLJN=YES`
   | `db2 "UPDATE DBM CFG USING query_heap_sz 32768"`
   | `db2 "UPDATE DBM CFG USING maxagents 500"`
   | `db2 "UPDATE DBM CFG USING sheapthres 0"`

4. **(Important)** Failure to complete this step will result in an unsuccessful database transfer. Run the following commands on the DB2 server system to create the necessary databases.

   **Notes:**
   - Replace `dbname` with the actual name of the database. Run the commands and each time replace `dbname` with the actual values for release, community, customization, Java Content Repository, Feedback, and Likeminds. You will need to run the commands once for each database for a total of six times. **Remember:** DB2 database names cannot
A. On the DB2 server system, run the following commands. This step is only required for the IBM Java Content Repository database (\texttt{jcrdb}).

- \texttt{jcrdb} is the name of the database used to store user data and objects

- \texttt{jcr} is the jcr user for \texttt{jcrdb} \textbf{Note:} This value can be replaced with any ID that has administrative authority.

- \texttt{dbpassword} is the password for the jcr user for the \texttt{jcrdb}

\机能\texttt{db2 "CONNECT TO jcrdb USER jcr USING dbpassword"}
\机能\texttt{db2 "CREATE BUFFERPOOL ICMLSFREQBP4 SIZE 1000 PAGESIZE 4 K"}
\机能\texttt{db2 "CREATE BUFFERPOOL ICMLSVOLATILEBP4 SIZE 16000 PAGESIZE 4 K"}
\机能\texttt{db2 "CREATE BUFFERPOOL ICMLSMAINBP32 SIZE 16000 PAGESIZE 32 K"}
\机能\texttt{db2 "CREATE BUFFERPOOL CMBMAIN4 SIZE 1000 PAGESIZE 4 K"}
\机能\texttt{db2 "CREATE REGULAR TABLESPACE ICMLFQ32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLFQ32') BUFFERPOOL ICMLSMAINBP32"}
\机能\texttt{db2 "CREATE REGULAR TABLESPACE ICMLNF32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLNF32') BUFFERPOOL ICMLSMAINBP32"}
\机能\texttt{db2 "CREATE REGULAR TABLESPACE ICMVFQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('ICMVFQ04') BUFFERPOOL ICMLSVOLATILEBP4"}
\机能\texttt{db2 "CREATE REGULAR TABLESPACE ICMSFQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('ICMSFQ04') BUFFERPOOL ICMLSFREQBP4"}
\机能\texttt{db2 "CREATE REGULAR TABLESPACE CMBINV04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('CMBINV04') BUFFERPOOL CMBMAIN4"}
\机能\texttt{db2 "CREATE SYSTEM TEMPORARY TABLESPACE ICMLSSYSTSPACE32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('icmlssystspace32') BUFFERPOOL ICMLSMAINBP32"}
\机能\texttt{db2 "CREATE SYSTEM TEMPORARY TABLESPACE ICMLSSYSTSPACE4 PAGESIZE 4 K MANAGED BY SYSTEM USING ('icmlssystspace4') BUFFERPOOL ICMLSVOLATILEBP4"}
\机能\texttt{db2 "DISCONNECT jcrdb"}
\机能\texttt{db2 "TERMINATE"}

B. For JDBC Type 2 connections only: On DB2 Connect, use a text editor to open the \texttt{/etc/services} file. If it does not specify the DB2 connection service port, add the following text to specify the port for the remote DB2 instance:

\机能\texttt{db2c_\_db2inst1port1/tcp # DB2 connection service port}

where \texttt{db2inst1} is the name of the DB2 instance ID on the system, and \texttt{port1} with the actual port number that is
assigned to the DB2 connection service in your DB2 server installation. The connection service port on the DB2
Client system, WebSphere Portal server, must match the connection service port on the DB2 server. The ports
should match by number but not necessarily by name.

C. Set up the correct service name by entering the following command on the DB2 server system:
   `db2 "UPDATE DBM CFG USING svcename svce_name"` where `svce_name` is the connection service port name that is specified above.

D. For JDBC Type 2 connections only: On DB2 Connect, set DB2COMM to TCP/IP by using the db2set command
   `db2set DB2COMM=tcpip`.

E. For JDBC Type 2 connections only: Catalog the TCP/IP node with the IP address of the remote database server on
   DB2 Connect:
   `db2 "catalog tcpip node remote_db_node_alias remote database_server_node server connection_service_port"` where:
   - `remote_db_node_alias` is the alias name of the database server that you are defining for the WebSphere Application
     Server node name. The alias name can contain one to eight characters.
   - `database_server_node` is the fully qualified host name of your database server system.
   - `connection_service_port` is the name of the DB2 connection service port that is configured in the `/etc/services`
     file on the database server system.

F. For JDBC Type 2 connections only, catalog the WebSphere Portal databases on DB2 Connect, where:
   - `remote_db_name_domain`, is the cataloged name of the databases on the server system for each domain.
   - `domain_alias_name`, is the database alias names that you are defining.
   - `remote_db_node_alias` is the name that was used previously when you cataloged the TCP/IP node in the previous
     step.
   The alias for each database must be different from the actual database name and can only contain up to eight
characters:
   `db2 "catalog db remote_db_name_release as release_alias_name at node remote_db_node_alias"
   db2 "catalog db remote_db_name_community as comm_alias_name at node remote_db_node_alias"
   db2 "catalog db remote_db_name_customization as cust_alias_name at node remote_db_node_alias"
   db2 "catalog db remote_db_name_fdbkdb as fdbkdb_alias_name at node remote_db_node_alias"
   db2 "catalog db remote_db_name_lmdb as lmdb_alias_name at node remote_db_node_alias"
   db2 "catalog db remote_db_name_jcrdb as jcrdb_alias_name at node remote_db_node_alias"

G. For JDBC Type 2 connections only: Log out of DB2 Connect by entering `db2 "terminate"`.

H. For JDBC Type 2 connections only, on DB2 Connect, test your remote connection by issuing the following command
   in the DB2 command window: `db2 "connect to alias_name user username using password"`, where `alias_name`
   is the alias name that you defined above, `username` is the database user, and `password` is the password assigned to
   the database user.

Parent topic: Configuring WebSphere Portal to use DB2
Previous topic: Create users
Next topic: Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- The page size of table spaces used by WebSphere Portal must be 8192 bytes.
- For details on creating table spaces refer to the documentation for the database.

**Prerequisites**
- Installing DB2
- Create users
- Creating remote databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`

   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds

3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword **IN** and a space. For example: `community.COMP_INST.tablespace=IN COMM8KSPACE`

   Repeat this step for each domain that you are transferring.

4. Save and close `dbdomain.space_mapping.properties`.

5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - **Windows**: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
   - **UNIX**: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

**Parent topic:** Configuring WebSphere Portal to use DB2  
**Previous topic:** Creating remote databases  
**Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:
  - `release.DbName=release`
  - `jcr.DbName=jcrdb`
  - `feedback.DbName=fdbkdb`
  - `likeminds.DbName=lmdb`
  - `community.DbName=commdb`
  - `customization.DbName=custdb`
- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace `dbdomain` with:
  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
- dbdomain.DbType
- dbdomain.DbName
- dbdomain.DbUrl
- dbdomain.DbSchema

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.
   A. For dbdomain.DbType, type db2.
   B. For dbdomain.DbName, type the name of the WebSphere Portal domain database. Notes:
      - This value is also the database element in the dbdomain.DbUrl property.
      - This value is the TCP-IP alias for the database.
   C. For dbdomain.DbSchema, type the schema name of the database domain. Note: Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   D. For dbdomain.DataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For dbdomain.DbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. Note: The database element of this value should match the value of DbName.
   F. For dbdomain.DbUser, type the user ID for the database administrator.
   G. For dbdomain.DbPassword, type the password for the database administrator.
   H. Optional: For dbdomain.DbRuntimeUser, type the user ID of the database user that should be used by WebSphere Portal to connect to the database at runtime.
   I. If dbdomain.DbRuntimeUser is specified, you must set dbdomain.DbRuntimePassword to be the password of the runtime database user.
   J. For dbdomain.DBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.
K. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.

L. For `dbdomain.XDbName`, type the database loop back alias that needs to be set if you plan to use the `create-database` task. **Note:** Required only for local databases using a JDBC Type 2 driver.

M. For `dbdomain.DbNode`, type the value for the node database. Set this value if you want to call `create-database`. **Note:** Required only for local databases.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For `source.domainDbType`, type of the database you are currently configured to use. The value for `source.domainDbType` is Derby by default.

B. For `source.domainName`, type the name of the database domain you are currently using.

C. For `source.domainSchema`, type current schema identifier for objects within the database for this domain.

D. For `source.domainDataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For `source.domainDbUrl`, type the url currently used to access your database.

F. For `source.domainDbUser`, type the name of the user accessing this database.

G. For `source.domainDbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

   A. For `db2.DbDriver`, type the name of the JDBC driver class.

   B. For `db2.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For `db2.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Configuring WebSphere Portal to use DB2  
**Previous topic:** Assigning custom table spaces  
**Next topic:** Configuring JCR collation support
Configuring JCR collation support

View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties

1. Stop the WebSphere Portal server:
   . ./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password

2. If using a remote DB2 server, copy the following files from the WebSphere Portal server to a temporary directory on the remote DB2 server:
   PortalServer_root/jcr/prereq.jcr/config/collation.jar
   PortalServer_root/jcr/prereq.jcr/config/registerCollationUDFTemplate.sql

3. Set up collation on the database where the JCR domain is located.
   A. Change to the directory db2_instance_owner_home/sqlib/function.
   B. Execute the command: Remote DB2:
db2 -tvf temporary location/collation.jar icm/CollationUDF.class
   Local DB2:
db2 -tvf PortalServer_root/jcr/prereq.jcr/config/collation.jar icm/CollationUDF.class
   C. If using a local DB2, change to the directory wp_profile_root/PortalServer/jcr/config.
   D. Open the file registerCollationUDFTemplate.sql and change all SCHEMA references to the JCR schema; for example, JCR. The value set for SCHEMA should match the value set for the jcr.DbSchema property. You specify jcr.DbSchema in the configuration file wkplc_comp.properties when you modify database properties.
   E. Connect to the JCR database by running db2 connect to <jcrdb> user <userid> using <password>.
   F. Execute the script by running the command Remote DB2:
db2 -tvf temporary location/registerCollationUDFTemplate.sql
   Local DB2:
db2 -tvf wp_profile_root/PortalServer/jcr/config/registerCollationUDFTemplate.sql
   G. Disconnect from the JCR database.
   H. Restart the DB2 instance.

4. Verify that the UDF is registered properly.
   A. Log in as the db2instanceID. Open a DB2 terminal window, and type db2. From the command line, type the command:
      connect to JCRDB user userid using password. You must connect to the JCR database using the userid and password of the WebSphere Portal JCR data source.
   B. Execute the following command in the previous substep. If the command completes successfully, the UDF is registered correctly as follows:
      values schema.sortkeyj('abc','en'), where schema is the schema used in the previous substep.

5. Edit the icm.properties file, located in wp_profile_root/PortalServer/jcr/lib/com/ibm/icm directory. Add the following section to the end of the file:
   # Enable/Disable collation support for all DB2 platforms
6. Enter the following command to start the WebSphere Portal server:

```
./startServer.sh WebSphere_Portal
```

Parent topic: Configuring WebSphere Portal to use DB2
Previous topic: Modifying database properties
Next topic: Configure WebSphere Portal to use DB2
Configure WebSphere Portal to use DB2

View information on how to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties

**Tips:**
- To run these tasks as a non-root user, you must first run the task chown -R *non-root_user* *WebSphereDir*.
- If you are transferring from Oracle, the `open_cursors` setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

**Steps for transferring data to another supported database**

1. If you are running a type 2 connection, before transferring data edit the `db2cli.ini` file. Failure to follow these steps will cause the database transfer to hang at the task `action-process-constraints`.
   A. Locate the file `/home/db2inst1/sqlib/cfg/db2cli.ini`.
   B. Add the following to the end of the file. Leave an empty line after `ReturnAliases=0.[COMMON]`
   ```
   DYNAMIC=1
   ReturnAliases=0
   ```

2. Perform this step only if you are installing multiple instances of WebSphere Portal. Change the maximum number of databases `MAX_NETBIOS_CONNECTIONS` to increase the default configured number of databases. For example, enter the following command at the database prompt: `set client MAX_NETBIOS_CONNECTIONS 254` A message indicates success if the number was increased.

3. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.

4. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2. 
   ```
   /home/db2inst1/sqlib/db2profile
   ```
   where `db2inst1` represents your database instance**Note:** You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.

5. Enter the following commands to validate configuration properties. 
   ```
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ```
./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password

6. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.

7. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td>./stopServer.sh server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

8. Transfer the database:

   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory wp_profile_root/ConfigEngine.

   B. Enter the following command: ./ConfigEngine.sh database-transfer -DWasPassword=password

      **Note:**
      - To select specific database domains to transfer, modify the -DTransferDomainList specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command: ./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
      - If you have been storing data in IBM® Derby for a long time, database transfer could fail with OutOfMemory exceptions. If database transfer fails, add the following property to the command in this step: ./ConfigEngine.sh database-transfer -DDbtJavaMaxMemory=1536M -DWasPassword=password

   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files and then repeat this step.

9. After transferring the database tables, perform a reorg check to improve performance. Perform this step for each database alias in the property file.

   A. Connect to a database with the following command:

   ```
   db2 connect to database_alias user db2admin_userid using password
   ```

   **Note:** Additional options might be required if additional security has been installed. Refer to DB2 Universal Database commands by example for links to the command reference.

   B. After it is connected, run the following commands from the DB2 prompt:

   ```
   db2 reorgchk update statistics on table all > xyz.out
   ```

   C. Look in the reorg column for entries marked with a * (star or asterisk) in the file xyz.out. For each line with a *, note the `tablename` and run the following command for each `tablename`:

   ```
   db2 reorg table tablename db2 terminate db2rbind
database_name -l db2rbind.out -u db2_admin -p password
   ```

   D. The output file db2rbind.out is only created when there is an error for the `db2rbind` command.

10. Change to the directory wp_profile_root/bin.

11. Enter the following command to start the WebSphere Portal server:

    ```
    ./startServer.sh WebSphere_Portal
    ```

**Parent topic:** Configuring WebSphere Portal to use DB2

**Previous topic:** Configuring JCR collation support

**Next topic:** Configuring DB2 for large file handling in Web Content Management

**Related information**

- [#SQL0443N with SYSIBM:CLI:-805 when invoking a CLI catalog function after upgrading to DB2 UDB Version 8.1 FixPak](#SQL0443N with SYSIBM:CLI:-805 when invoking a CLI catalog function after upgrading to DB2 UDB Version 8.1 FixPak)
Configuring DB2 for large file handling in Web Content Management

If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties
  - Configure WebSphere Portal to use DB2

  **Note:** You only need to perform these steps if you are using Web Content Management.

1. Log into the WebSphere Application Server administrative console.
2. Click **Resources > JDBC > Data sources**.
3. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
4. Select the name of the data source that is defined in `wkplc_comp.properties` for the JCR database domain. The default data source is `wpdbDS`.
5. Click **Custom properties**.
6. Ensure that the `fullyMaterializeLobData` property is set to `false`. 

**Parent topic:** Configuring WebSphere Portal to use DB2

**Previous topic:** Configure WebSphere Portal to use DB2

**Next topic:** Changing driver types
Changing driver types

WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Before you begin, ensure that the following conditions are met:
- The WebSphere Portal database has been successfully transferred to DB2 using the database-transfer configuration task.
- The files wkplc_comp.properties and wkplc_dbtype.properties have been modified to set the correct values for the DB2 drivers that you are switching to:

  In the file wkplc_comp.properties set each `<Domain>.DbUrl` property using the following formats:
  
  # db2 (type 2):       { jdbc:db2:wpsdb }
  # db2 (type 4):       { jdbc:db2://<YourDatabaseServer>:50000/wpsdb:returnAlias=0; }

  In the file wkplc_dbtype.properties set the `db2.DbLibrary` property using the following format:
  
  # For DB2 Type 2 driver use `<SQLLIB>/java/db2java.zip`
  # For DB2 Type 4 driver use `<SQLLIB>/java/db2jcc.jar:<SQLLIB>/java/db2jcc_license_cu.jar`

  In the file wkplc_dbtype.properties set the `db2.DbDriver` property using the following format:
  
  # For DB2 Type 2 driver use COM.ibm.db2.jdbc.app.DB2Driver
  # For DB2 Type 4 driver use com.ibm.db2.jcc.DB2Driver

**Note:** If WebSphere Portal is installed on the same machine as the DB2 server and you switch from a JDBC Type 4 connection to a JDBC Type 2 connection, verify that you have created the alias names for the DB2 databases as described in Creating remote databases and that the alias names are specified for the databases in the file wkplc_comp.properties. When switching from a JDBC Type 2 connection to a JDBC Type 4 connection, remove the database alias names and refer to the databases directly. This is required because of a limitation in the DB2 Universal JDBC driver.

**Prerequisites**
- Installing DB2
- Create users
- Creating remote databases
- Modifying database properties
- Configure WebSphere Portal to use DB2
- Configuring DB2 for large file handling in Web Content Management

1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`. 
2. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2. 

   ```bash
   /home/db2inst1/sqllib/db2profile
   ```

   where `db2inst1` represents your database instance

   **Note:** You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.

3. Enter the following commands to validate configuration properties:

   ```bash
   DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ```
4. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td><code>.stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td><code>.stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

6. Change to the directory `wp_profile_root/ConfigEngine`.

7. To change from one supported driver to the other, run the following task to connect the database, including only the domains that require the switch. `.ConfigEngine.sh connect-database -Drelease.DbPassword=password -Dcustomization.DbPassword=password -Dcommunity.DbPassword=password -Djcr.DbPassword=password -Dfeedback.DbPassword=password -Dlikeminds.DbPassword=password -DWasPassword=password`

8. Change to the directory `wp_profile_root/bin`.

9. Enter the following command to start the WebSphere Portal server: `.startServer.sh WebSphere_Portal`

Parent topic: Configuring WebSphere Portal to use DB2
Previous topic: Configuring DB2 for large file handling in Web Content Management
Configuring WebSphere Portal to use DB2 for z/OS

View information on installing and setting up DB2 for z/OS to work with WebSphere Portal.

1. Installing DB2 for z/OS
   View information on how to install DB2 for z/OS for use with IBM® WebSphere® Portal.

2. Creating users
   Complete this procedure to set up users for IBM DB2 Universal Database™ for z/OS® to work with IBM WebSphere Portal.

3. Creating remote databases
   A remote database resides on a different machine than WebSphere Portal. You must manually create the required databases before configuring WebSphere Portal to work with DB2 for z/OS.

4. Optional: Creating the Java Content Repository database
   View information on setting up your Java Content Repository database to work with WebSphere Portal.

5. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

6. Configuring WebSphere Portal to use DB2 for z/OS
   View information on manually transferring data to the DB2 for z/OS you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository to DB2 for z/OS. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configure WebSphere Portal to use a remote database
Installing DB2 for z/OS

View information on how to install DB2 for z/OS for use with IBM® WebSphere® Portal.

The following prerequisites must exist separately from the WebSphere Portal installation:

- If you are using the DB2 Legacy Type 2 JDBC driver, configure your DB2 Connect client with the following commands:
  - `db2 update dbm cfg using tp_mon_name WAS`
  - `db2 update dbm cfg using spm_name hostname`, where `hostname` is the host name for the machine where the DB2 Connect client is installed (same as portal machine).

- If you are using the DB2 Legacy Type 2 JDBC driver, enable extended shared memory usage with the following commands:
  - `export EXTSHM=ON`
  - `db2set DB2ENVLIST=EXTSHM`
  - `db2start`

  To make the change permanent, you must add the environment variable by adding the following to the `profile.env` file:
  ```
  DB2ENVLIST='EXTSHM'
  in /home/db2inst/sqllib/userprofile add: export EXTSHM=ON
  ```

  **Note:** The shell must be reopened before restarting DB2 for z/OS.

- Both type 2 and type 4 drivers are supported. If you are using the DB2 Legacy Type 2 JDBC driver, the DB2 Connect client must be installed on the same machine as WebSphere Portal to connect to the remote database.

- The DB2 subsystem must be on a supported z/OS platform.

- Update the BP8K0 catalog bufferpool to 35,000 buffers before performing database transfer. You should change this value based on your environment. The SYSIBM.SYSDATABASE table resides in this bufferpool and is used extensively by DB2 for z/OS during database transfer.

To use DB2 for z/OS as the database software for WebSphere Portal, you must have DB2 for z/OS installed on your z/OS system. Refer to the DB2 for z/OS product documentation for instructions. Refer to the Planning for DB2 for z/OS topic for additional considerations for configuring DB2 for z/OS as the WebSphere Portal database.

**Parent topic:** Configuring WebSphere Portal to use DB2 for z/OS

**Next topic:** Creating users
Creating users

Complete this procedure to set up users for IBM® DB2 Universal Database™ for z/OS® to work with IBM WebSphere® Portal.

Before you begin: You should have completed Installing DB2 for z/OS.

- **Prerequisites**
  - Installing DB2 for z/OS

- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

- If you are configuring multiple WebSphere Portal instances to use a single DB2 for z/OS subsystem, you must create a unique database user for each WebSphere Portal instance. By default all database table names include the name of the database user used to access the data. Therefore, to prevent table name conflicts, you must create a unique database user for each WebSphere Portal instance on the shared DB2 for z/OS subsystem.

- Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Use the following steps to grant access permissions to the database users. Repeat these steps for each WebSphere Portal instance you are setting up.

1. Create all database user IDs in the security product you are using on z/OS. For `jcrschema`, the database schema name for IBM Java Content Repository data, create a group and connect it to the database user ID for Java Content Repository data, `jcr`. The following sample shows the RACF definition of such a user ID and group, where `jcr` is the database user ID for Java Content Repository data, `yourDefaultUserGroup` is your default RACF group for database user IDs, `jcrschema` is the database schema name for Java Content Repository data, and `yourDefaultGroup` is your default RACF group for groups. If you have some other security product such as Top Secret or ACF2 instead of RACF, translate the sample RACF definition into the appropriate syntax before running the job.

   ```
   ADDUSER jcr DFLTGRP(yourDefaultUserGroup) NAME('WAS DB2 ACCESS USER')
   PW USER(jcr) NOINTERVAL
   ALU jcr PASSWORD(*******) NOEXPIRED
   ADDGROUP jcrschema SUPGROUP(yourDefaultGroup)
   CONNECT jcr GROUP(jcrschema)
   ```

2. Run the following SQL statements using a tool like SPUFI while logged on to the DB2 subsystem to grant appropriate rights on the newly created databases. If you are configuring multiple WebSphere Portal instances to use a single DB2 for z/OS subsystem, be sure to use the database user associated with the database user ID for Java Content Repository data, `jcr`. The following SQL statements show the rights to be granted for each set of database user IDs.

   ```
   (C) create/alter tablespaces
   (C) create/alter tables
   (C) create/alter indice;
   (C+R) read/write data
   (C) - at configuration time
   (R) - at runtime
   ```
GRANT USE OF ALL BUFFERPOOLS TO releaseusr;
GRANT USE OF ALL BUFFERPOOLS TO communityusr;
GRANT USE OF ALL BUFFERPOOLS TO customizationusr;

GRANT DBADM ON DATABASE jcrdbnameonzos TO jcr;
GRANT USE OF ALL BUFFERPOOLS TO jcr;
GRANT DBADM ON DATABASE fdbkdbnameonzos TO feedback;
GRANT USE OF ALL BUFFERPOOLS TO feedback;
GRANT DBADM ON DATABASE lmddbnameonzos TO lmdbusr;
GRANT USE OF ALL BUFFERPOOLS TO lmdbusr;
GRANT SELECT ON SYSCOLUMNS TO releaseusr;
GRANT SELECT ON SYSCOLUMNS TO communityusr;
GRANT SELECT ON SYSCOLUMNS TO customizationusr;
GRANT SELECT ON SYSTABLES TO releaseusr;
GRANT SELECT ON SYSTABLES TO communityusr;
GRANT SELECT ON SYSTABLES TO customizationusr;
GRANT SELECT ON SYSCOLUMNS TO jcr;
GRANT SELECT ON SYSTABLES TO jcr;
GRANT SELECT ON SYSCOLUMNS TO fdbkbusr;
GRANT SELECT ON SYSTABLES TO fdbkbusr;
GRANT SELECT ON SYSCOLUMNS TO lmdbusr;
GRANT SELECT ON SYSTABLES TO lmdbusr;
GRANT SELECT ON SYSFOREIGNKEYS TO releaseusr;
GRANT SELECT ON SYSFOREIGNKEYS TO communityusr;
GRANT SELECT ON SYSFOREIGNKEYS TO customizationusr;
GRANT SELECT ON SYSFOREIGNKEYS TO jcr;
GRANT SELECT ON SYSFOREIGNKEYS TO fdbkbusr;
GRANT SELECT ON SYSFOREIGNKEYS TO lmdbusr;
GRANT SELECT ON SYSREL TO releaseusr;
GRANT SELECT ON SYSREL TO communityusr;
GRANT SELECT ON SYSREL TO customizationusr;
GRANT SELECT ON SYSREL TO jcr;
GRANT SELECT ON SYSREL TO fdbkbusr;
GRANT SELECT ON SYSREL TO lmdbusr;
GRANT USE OF STOGROUP jcrstogroup TO jcr;
GRANT CREATEIN, DROPIN ON SCHEMA jcrschema TO jcr;
GRANT SELECT ON SYSTABLESPACE TO jcr;
GRANT SELECT ON SYSVIEWS TO jcr;
GRANT SELECT ON SYSDUMMY1 TO jcr;
GRANT SELECT ON SYSTRIGGERS TO jcr;
GRANT SELECT ON SYINDEXPART TO jcr;
GRANT SELECT ON SYINDEXES TO jcr;
GRANT SELECT ON SYSYNONYMS TO jcr;

where:
- `releasenameonzos`, `communitynameonzos`, `customizationnameonzos`, and `releaseusr`, `communityusr`, `customizationusr` represent the databases and database users, respectively, of the WebSphere Portal instance you are setting up. (These users must be created on the host system.)
- `jcrdbnameonzos` and `jcr` are the database and database user, respectively, for Content Repository data.
- `fdbkdbnameonzos` and `feedback` are the database and database user, respectively, for Feedback data.
- `lmddbnameonzos` and `lmdbusr` are the database and database user, respectively, for Likeminds data.
- `jcrschema` is the database schema name for Content Repository data.

3. Grant the necessary access rights to all users who might require them. Depending on the architecture that you choose, these users might include Java Content Repository and Feedback users.

**Parent topic:** Configuring WebSphere Portal to use DB2 for z/OS  
**Previous topic:** Installing DB2 for z/OS  
**Next topic:** Creating remote databases
Creating remote databases

A remote database resides on a different machine than WebSphere Portal. You must manually create the required databases before configuring WebSphere Portal to work with DB2 for z/OS.

Before you begin, complete the following prerequisites:

- These instructions assume that you will use a local IBM® DB2 Universal Database™ Enterprise Server Edition Connect to connect to a remote IBM DB2 Universal Database for z/OS® server, and that IBM WebSphere® Application Server, WebSphere Portal, and DB2 Connect will be installed on the same machine.
- To run SQL statements, you can use a tool like SPUFI. The specified statements include CREATE STOGROUP, CREATE DATABASE and CREATE TABLESPACE. For CREATE STOGROUP you have to replace the icmvolumes and icmvcat variables with volume serial numbers and a catalog name for your environment. For CREATE TABLESPACE you can specify a specific BUFFERPOOL. Refer to the DB2 for z/OS SQL Reference for a description of these options.
- Ensure that a TEMP database is created for your subsystem. You can use the following statements to create a TEMP database:

```
CREATE DATABASE db_name AS TEMP;
CREATE TABLESPACE ts_name IN db_name;
```

- Log on to the DB2 subsystem on the host server. DB2 system administrator rights are needed to create the databases.
- Replace variables as follows:
  - releasenameonzos, communitynameonzos, and customizationnameonzos are the WebSphere Portal databases for WebSphere Portal and Member Manager data.
  - fdbkdbnameonzos and fdbkbts are the database and table space, respectively, for Feedback data.
  - lmdbnameonzos and lmdbts are the database and table space, respectively, for LikeMinds data.
- Because large objects are stored in columns that can become very large, logging changes to these columns requires a huge amount of log space. For this reason, large object (LOB) logging is disabled by default for table spaces that contain such data. With LOB logging disabled, you can recover full backups, but not incremental backups that can be used for point in time recovery. To recover point in time backups, you must enable LOB logging. For detailed instructions, see Technote 1306637, Managing LOB logging in DB2 for z/OS.

**Prerequisites**

- Installing DB2 for z/OS
- Creating users

Use the following steps to set up WebSphere Portal with a remote instance of DB2 for z/OS. Refer to Planning for DB2 for z/OS for a list of databases and table space names. If you are configuring multiple WebSphere Portal instances to use a single DB2 subsystem, be sure to use the database and table space names associated with the WebSphere Portal instance you are setting up.

1. `CREATE DATABASE releasenameonzos CCSID UNICODE;`
2. `CREATE DATABASE communitynameonzos CCSID UNICODE;`
3. `CREATE DATABASE customizationnameonzos CCSID UNICODE;`
4. Execute the steps in the topic Creating the Java Content Repository database.
5. `CREATE DATABASE fdbkdbnameonzos CCSID UNICODE;`
6. `CREATE TABLESPACE fdbkdbts IN fdbkdbnameonzos USING STOGROUP SYSDEFLT PRIQTY 5000 SECQTY 500 LOCKSIZE ROW;`
7. `CREATE DATABASE lmdbnameonzos CCSID UNICODE;`
8. CREATE TABLESPACE lmdbts IN lmdbnameonzos USING STOGROUP SYSDEFLT PRIQTY 5000 SECQTY 500;

Parent topic: Configuring WebSphere Portal to use DB2 for z/OS
Previous topic: Creating users
Next topic: Creating the Java Content Repository database

Related Information
- Managing LOB logging in DB2 for z/OS
Creating the Java Content Repository database

View information on setting up your Java Content Repository database to work with WebSphere Portal.

First, you need to create multiple databases for scalability.

The IBM Java Content Repository database grows in size as the IBM® WebSphere® Portal server is used. To support this growth, which includes the dynamic creation of many new tables within the database, the WebSphere Portal server allows you to create multiple databases within a single IBM DB2 Universal Database™ for z/OS® subsystem. A minimum of two databases is recommended, but more can be created if necessary. A single database setup is not recommended; it will quickly become constrained. See the Performance guides on the product documentation Web site for more information.

Each database that is created must begin with a common prefix; there is no convention required for the remainder of the name. For example, to create \( \times x \) number of databases, you may choose to use the following commands:

```sql
CREATE DATABASE JCRDB01
CREATE DATABASE JCRDB02
...
CREATE DATABASE JCRDBXX
```

In this case, \( JCR \) is the common prefix. You also have to select a maximum number of user-defined tables (UDT) that each database will be allowed to hold; the recommended value is 400.

**- Prerequisites**

- Installing DB2 for z/OS
- Creating users
- Creating remote databases

A sample script of the commands that you can use to create the Java Content Repository database in your DB2 for z/OS installation is shown below. The sample demonstrates the creation of a single database. To follow the recommendation of creating at least two databases, duplicate the commands for each additional database as indicated below. Find a template of these commands in the file `PortalServer_root installer/wp.config/config/templates/db/db2_zos/jcr_sample.sql`.

**Notes:**

- It is not necessary to duplicate every tablespace definition in every database.
- In order to run the commands shown in the sample, you must know the database name and the IDs of the MVS volumes where the Java Content Repository database tables will be stored.
- Edit the template file for your installation environment before running the commands shown in the sample:
  - Replace `jcrdbnameX` with the name of the database. Remember that each database name must begin with a common prefix.
  - Replace `stogroup` with the name of your storage group.
  - Replace `icmvolumes` with the IDs of the MVS volumes where the Java Content Repository database tables will be stored. These values are assigned by the database administrator.
  - Replace `icmcatalog` with the name of the virtual catalog.
  - Replace `jcr` with the name of database user ID.
  - Replace `4kbp` with the name of your 4K bufferpool.
  - Replace `32kbp` with the name of your 32K bufferpool.
  - Replace `jcrschema` with the schema name of your Java Content Repository domain.
--DROP DATABASE jcrdbnameX
--DROP STOGROUP stogroup
--DROP ALIAS jcrschema.ICMFICTITIOUS

CREATE STOGROUP stogroup VOLUMES (icmvolumes) VCAT icmvcat
GRANT USE OF STOGROUP stogroup to jcr

CREATE DATABASE jcrdbnameX STOGROUP stogroup BUFFERPOOL 4kbp INDEXBP 4kbp CCSID UNICODE

Note: The following two tablespace commands must be created in every database:
CREATE TABLESPACE ICMLFQ32 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICMTSQ04 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

Note: The following tablespaces are required in the first database only:
CREATE TABLESPACE ICMLFQ32 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICMTSQ04 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 5000 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSTADO IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRIDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRSCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRISLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRGCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRIGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSTDPV IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSDOAC IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RESMLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COLNLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USERLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USEGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE SYSCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CACLLSTS IN jcrdbnameX USING STOGROUP storgroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CPERLSTS IN jcrdbnameX USING STOGROUP storgroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTGLSTS IN jcrdbnameX USING STOGROUP storgroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTDLSTS IN jcrdbnameX USING STOGROUP storgroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CPERLSTS IN jcrdbnameX USING STOGROUP storgroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTDLLSTS IN jcrdbnameX USING STOGROUP storgroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTDLSTS IN jcrdbnameX USING STOGROUP storgroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE ICUT301 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE ICMACLS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE ICMACLC IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE ICMACTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00208 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00207 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00206 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00205 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00204 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00203 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00210 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00202 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00209 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00201 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00200 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TIELLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE REPLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE REPRLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE RI11LSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE XDOFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE XDOOLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE IDELLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TEICLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE SYAELSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

1154
CREATE TABLESPACE TSINJRTS IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TSERJRTS IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TIMEJRTS IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE SPRTJRTS IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RWSJRTS IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NODEJRTS IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ROOTJRTS IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRPDJRTS IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NSURJRTS IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NSPRJRTS IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NODRJRTS IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NODTJRTS IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NODDJRTS IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE MAXSJRTS IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE LINKJRTS IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE GLBPJRTS IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE DWSLJRTS IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ADDPJRTS IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT601 IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT341 IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT331 IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT321 IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT311 IN LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TSPEJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TSTAJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RIDSJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE WSN0JRTS IN jcrdbnameX USING STOGROUP stogroupPRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE LOB TABLESPACE ICMLOBT1 IN jcrdbnameX LOG NO
CREATE LOB TABLESPACE ICMLOBT2 IN jcrdbnameX LOG NO
CREATE LOB TABLESPACE ICMLOBT3 IN jcrdbnameX LOG NO
CREATE ALIAS jcrschema.ICMFICTITIOUS FOR SYSIBM.SYSDUMMY1
CREATE TABLESPACE DLKRJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE LKRLJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

Parent topic: Configuring WebSphere Portal to use DB2 for z/OS
Previous topic: Creating remote databases
Next topic: Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root` /PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.
- When using IBM® DB2 Universal Database™ for z/OS® as a data store, WebSphere Portal requires that its indexes are not padded. Therefore, you must set the DSNZPARM parameters to `RETVLCFK=NO` or `PADIX=NO`, or both.

- **Prerequisites**
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.

2. Open the mapping file `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`

   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds

3. Assign a table space to each `.tablespace` entry in the mapping file. Assignments to `.indexspace` entries are ignored.

   The table space name must be qualified by the database name and prepended by the keyword `IN` and a space. For example:`community.COMP_INST.tablespace=IN COMM8KSPACE`

   Repeat this step for each domain that you are transferring.

4. Save and close `dbdomain.space_mapping.properties`.

5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - **Windows:** ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
- UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

Parent topic: Configuring WebSphere Portal to use DB2 for z/OS
Previous topic: Creating the Java Content Repository database
Next topic: Configuring WebSphere Portal to use DB2 for z/OS
Configuring WebSphere Portal to use DB2 for z/OS

View information on manually transferring data to the DB2 for z/OS you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository to DB2 for z/OS. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

- Prerequisites
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases

Steps for transferring data to another supported database

1. If you are running a type 2 connection, before transferring data edit the db2cli.ini file. Failure to follow these steps will cause the database transfer to hang at the task action-process-constraints.
   A. Locate the file /home/db2inst1/sqllib/cfg/db2cli.ini.
   B. Add the following to the end of the file. Leave an empty line after ReturnAliases=0.[COMMON]

   DYNAMIC=1
   ReturnAliases=0

2. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.

3. Enter the following commands to validate configuration properties...
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password

4. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td>./stopServer.sh server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

6. Transfer the database: Note: Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.
   A. Change to the directory wp_profile_root/ConfigEngine.
   B. Enter the following command:.

   ./.ConfigEngine.sh database-transfer -DWasPassword=password

   Note: To select specific database domains to transfer, modify the -DTransferDomainList specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter
the following command: 

```
./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
```

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files and then repeat this step.

7. From the command line processor, reset the check pending status on the WebSphere Portal table spaces listed here.

CHECK DATA is a DB2 batch utility that is invoked through JCL or through the DB2 utility panels. Type the following utility commands to check pending status:

```
check data tablespace releasenameonzos.TS320A
check data tablespace releasenameonzos.TS280A
check data tablespace releasenameonzos.TS300A
check data tablespace releasenameonzos.TS2110A
check data tablespace releasenameonzos.TS830A
check data tablespace communitynameonzos.TS8000B
check data tablespace communitynameonzos.TS8011B
check data tablespace communitynameonzos.TS280B
check data tablespace customizationnameonzos.TS2110C
check data tablespace jcrdbnameonzos.ICMSFQ04
check data tablespace jcrdbnameonzos.TS2111D
```

where releasenameonzos, communitynameonzos, and customizationnameonzos are the names of your WebSphere Portal databases, and jcrdbnameonzos is the name of your JCR database. Refer to the Utility Guide and Reference for DB2 for z/OS for additional details.

8. Run the RUNSTATS utility as shown in the following example:

```
LISTDEF JCRDBZOS INCLUDE TABLESPACE JCRDBZOS.* BASE
RUNSTATS TABLESPACE LIST JCRDBZOS INDEX(ALL) KEYCARD TABLE(ALL)
LISTDEF JCRDB01 INCLUDE TABLESPACE JCRDB01.* BASE
RUNSTATS TABLESPACE LIST JCRDB01 INDEX(ALL) KEYCARD TABLE(ALL)
...```

9. Start the WebSphere Application Server.

10. Verify that the WebSphere Portal application server is running by opening the following URL in a browser:

```
http://hostname.companyname.com:port_number/wps/portal
```

where hostname.companyname.com is the fully qualified host name of the machine where WebSphere Portal is running and port_number is the transport port that is created by WebSphere Application Server.

If you have performed database transfer in a clustered environment, you must explicitly copy the wp_profile_root/PortalServer/jcr/lib/com/ibm/icm/icm.properties file from the primary node on which the database-transfer task was processed to all secondary nodes. This ensures that the secondary nodes have the new JCR database values specified in the icm.properties file.

1. Stop the portal server on the secondary node.
2. Copy the wp_profile_root/PortalServer/jcr/lib/com/ibm/icm/icm.properties file from the primary node and replace the icm.properties file on the secondary node with the new file from the primary node.
3. Start the portal server on the secondary node.

Parent topic: Configuring WebSphere Portal to use DB2 for z/OS
Previous topic: Assigning custom table spaces
Configuring WebSphere Portal to use Oracle

View information on installing and setting up Oracle to work with WebSphere Portal.

1. Installing Oracle
   View information on how to install Oracle for use with WebSphere Portal.

2. Creating users
   View the steps to set up users for Oracle to work with WebSphere Portal.

3. Creating databases
   View some important considerations before setting up Oracle databases to work with WebSphere Portal.

4. Modifying database properties
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

6. Creating JCR table spaces
   This topic provides manual instructions for creating JCR table spaces for Oracle.

7. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

8. Configure WebSphere Portal to use Oracle
   This section provides information on how to manually transfer data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

**Parent topic:** Configure WebSphere Portal to use a remote database
Installing Oracle

View information on how to install Oracle for use with WebSphere Portal.

You can use Oracle as the database software. You must install Oracle separately and before performing a database transfer.

Refer to Oracle product documentation for installation instructions.

Parent topic: Configuring WebSphere Portal to use Oracle
Next topic: Creating users
Creating users

View the steps to set up users for Oracle to work with WebSphere Portal.

Before you begin:

- You should have installed Oracle.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Start the Oracle SQL command line processor ‘SQL*Plus’ by entering `sqlplus /nolog` on the operating system command prompt.

2. Log in to the Oracle database by executing command `connect user/password@database name`. For example: `system/manager@wpsdb` will log the administrative user system with a password of manager into the wpsdb database.

3. Create the WebSphere Portal user `dbname.DbUser`, where `dbname` is replaced by release, community, or customization.

   A. Log in to the database in which you want to create the new users.

   B. Create the users that you need in the appropriate databases by entering the following commands in the SQL*Plus tool, where `user_tablespace` is the default tablespace that is identified by the database administrator to store user objects and `temp_tablespace` is identified to store temporary objects. All names must be in lower case.

   ```sql
   SQL> create user releaseusr identified by password
   default tablespace user_tablespace
   temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO releaseusr;
   SQL> create user communityusr identified by password
   default tablespace user_tablespace temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO communityusr;
   SQL> create user customizationusr identified by password
   default tablespace user_tablespace
   temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO customizationusr;
   SQL> create user jcr identified by password
   default tablespace user_tablespace
   temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO jcr
   ```

4. Connect to the Feedback database:

   A. Enter the following command: `SQL> connect`.

   B. Enter `user-name: username/password@dbname` where `username` is an existing administrative user in the database. For example, `system/manager@fdbkdb` will log the administrative user system with a password of manager into the fdbkdb database.
C. Create the Feedback user:

```
SQL> create user feedback identified by password
           default tablespace user_tablespace
           temporary tablespace temp_tablespace;
```

GRANT UNLIMITED TABLESPACE TO feedback

D. Log out of the command line tool using the command SQL> exit.

5. Connect to the LikeMinds database:

A. Enter the following command:

```
SQL> connect
```

B. Enter 

```
user-name: username/password@dbname, where username is an existing administrative user in the database.
```

For example, system/manager@lmdb will log the administrative user system with a password of manager into the

lmdb database.

C. Create the LikeMinds user:

```
SQL> create user lmdbusr identified by password
           default tablespace user_tablespace
           temporary tablespace temp_tablespace;
```

GRANT UNLIMITED TABLESPACE TO lmdbusr

D. Log out of the command line tool using the command SQL> exit.

6. Perform the following steps to verify that you can connect to the database with the user ID and password that you just created:

A. Enter the SQL> connect command to connect to the content database.

B. Enter 

```
user-name: username/password@dbname, where username is an existing administrative user in the database.
```

For example, system/manager@jcrdb will log the administrative user system with a password of manager into the Oracle
database.
Creating databases

View some important considerations before setting up Oracle databases to work with WebSphere Portal. For information about creating databases, refer to the Oracle product documentation. For information on the recommended database architecture and the databases you will need to create, see the Planning for Oracle topic. Be sure that all databases to be used with WebSphere Portal are created as UNICODE character set databases.

If you are using remote Oracle 9 or 10 databases, you must also copy the ojdbc14.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM® WebSphere® Application Server Version 7.0, you must also copy the ojdbc6.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM WebSphere Application Server Version 6.1, you must also copy the ojdbc5.jar file from the remote Oracle server to the WebSphere Portal machine. The typical location is the oracle_home/sqldeveloper/jdbc/lib directory. Record the copy location on your local machine for future reference.

When creating Oracle databases for use with WebSphere Portal, you should consider the following information:

- The Oracle databases must be created manually before configuring WebSphere Portal.
- All databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle server will be remote or local to the WebSphere Portal installation.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle database in order for WebSphere Portal to communicate with the Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes:

  - **db_block_size** = 8192 bytes
  - **db_cache_size** = 307,200 bytes
  - **db_files** = 1024 files
  - **log_buffer** = 65536 bytes
  - **open_cursors** = 1500 cursors
  - **pga_aggregate_target** = 204,800 bytes
  - **pre_page_sga** = true
  - **processes** = 300 processes
  - **shared_pool_size** = 204,800 bytes

**Note:** If you are using IBM Java Content Repository, the open_cursors value may need to be increased based on the table count in the Java Content Repository schema.

- Raise the number of parallel servers as appropriate. For example, if you have more than 875 parallel servers, you should set the parallel_max_servers to 1200.
- The Oracle parameter CURSOR_SHARING allows similar SQL Statements to be shared when possible, which prevents parsing and establishing a new execution plan. The execution plan is used by Oracle to gather the data needed to satisfy a request. There are two options for CURSOR_SHARING, which are as follows:

  - **FORCE**
    - When you select this option, Oracle uses the same execution plan for all SQLs that are similar in value even if the values are different. When you use this option, the execution plan may not provide optimum performance. For example, similar SQLs with different values may behave differently when executed running the same plan.

  - **EXACT**
- When you select this option, Oracle only shares the same execution plan for SQLs that are identical and use the same values. This option removes the risk of a SQL statement being executed when optimum performance conditions do not exist.

WebSphere Portal supports both options. Regardless of the option selected, portlet applications should not be affected. Contact your database administrator for further assistance on these options.

- **Prerequisites**
  - Creating users

**Parent topic:** Configuring WebSphere Portal to use Oracle

**Previous topic:** Creating users

**Next topic:** Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Creating users
  - Creating databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:

- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**

- The WebSphere Portal database can be used to hold information for applications such as Feedback and Like Minds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:

  - `release.DbName=release`
  - `jcr.DbName=jcrdb`
  - `feedback.DbName=fdbkdb`
  - `likeminds.DbName=lmdb`
  - `community.DbName=commdb`
  - `customization.DbName=custdb`

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.

- Depending on which database domain has to be configured, replace `dbdomain` with:

  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`

- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:

  - `ddomain.DbType`
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.

   A. For dbdomainDbType, type oracle.

   B. For dbdomainDbName, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the dbdomainDbUrl property.

   C. For dbdomainDbSchema, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The dbdomainDbName should be the same value used for the dbdomainDbSchema.

      **Restriction:** The value for dbdomainDbSchema must equal the value for dbdomainDbUser.

   D. For dbdomainDataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback

   E. For dbdomainDbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:**
      - The database element of this value should match the value of DbName.
      - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following:
        jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME. **When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.**

   F. For dbdomainDbUser, type the user ID for the database administrator. **Restriction:** The value for dbdomainDbUser must equal the value for dbdomainDbSchema.
For **dbdomain**.DbPassword, type the password for the database administrator.

For **dbdomain**.DBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.

For **dbdomain**.DBA.DbPassword, type the database administrator password for privileged access operations during creation of the database.

For **dbdomain**.DbHome, type the root location for the database. **Note:** This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For **source.domain**.DbType, type of the database you are currently configured to use. The value for **source.domain**.DbType is Derby by default.

B. For **source.domain**.DbName, type the name of the database domain you are currently using.

C. For **source.domain**.DbSchema, type current schema identifier for objects within the database for this domain.

D. For **source.domain**.DataSourceName, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For **source.domain**.DbUrl, type the url currently used to access your database.

F. For **source.domain**.DbUser, type the name of the user accessing this database.

G. For **source.domain**.DbPassword, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file wkplc_dbtype.properties.

A. For oracle.DriverManager, type the name of the class that SqlProcessor uses to import SQL files.

B. For oracle.DbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

C. For oracle.JdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file wkplc.properties.

A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Configuring WebSphere Portal to use Oracle  
**Previous topic:** Creating databases  
**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

- **Prerequisites**
  - Creating users
  - Creating databases
  - Modifying database properties

1. On the database server, make sure that the folder `db_1` contains the subfolders `your_oracle_instance/data` and `your_oracle_instance/index`. If this folder hierarchy does not exist, create it manually before you run the `setup-database` task. The `setup-database` task requires these folders to create database users. If these folders do not exist, the `setup-database` task will fail.

2. Change to the directory `wp_profile_root/ConfigEngine`

3. To create the database users, type the following command:

   ```bash
   ./ConfigEngine.sh setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:

   `\<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql`

**Parent topic:** Configuring WebSphere Portal to use Oracle

**Previous topic:** Modifying database properties

**Next topic:** Creating JCR table spaces

**Related tasks**

Manually creating users and granting privileges for Oracle
Creating JCR table spaces

This topic provides manual instructions for creating JCR table spaces for Oracle.

- Prerequisites
  - Creating users
  - Creating databases
  - Modifying database properties
  - Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.

2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:

   - `&jcrdb` is the name of the database you created to store user data.
   - `&dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.

   Ensure that the `.` is included in the variables when you substitute the values of your environment with these variables.

   **Important:** You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

   ```sql
   create tablespace ICMLFQ32 datafile '&dbpath./&jcrdb./data/ICMLFQ32_01.dbf' size 300M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLNF32 datafile '&dbpath./&jcrdb./data/ICMLNF32_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMVFQ04 datafile '&dbpath./&jcrdb./data/ICMVFQ04_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMSFQ04 datafile '&dbpath./&jcrdb./data/ICMSFQ04_01.dbf' size 150M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLSNDX datafile '&dbpath./&jcrdb./index/ICMLSNDX_01.dbf' size 10M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   ```

   A. Set the `size`, `autoextend`, and `maxsize` values according to your environment. For example, you may want to change the `maxsize` to a set value rather than `UNLIMITED`.

   B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.

   C. Refer to the Oracle command reference for more information about using the `create tablespace` command.

Parent topic: Configuring WebSphere Portal to use Oracle
Previous topic: Setting up databases
Next topic: Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:

- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

- Prerequisites
  - Creating users
  - Creating databases
  - Modifying database properties
  - Setting up databases
  - Creating JCR table spaces

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.index_name.indexspace
   For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
     - release
     - community
     - customization
     - jcr
     - feedback
     - likeminds
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword TABLESPACE and a space. For example: community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE
   Repeat this step for each domain that you are transferring.
4. Save and close dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
- UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

**Parent topic:** Configuring WebSphere Portal to use Oracle  
**Previous topic:** Creating JCR table spaces  
**Next topic:** Configure WebSphere Portal to use Oracle
Configure WebSphere Portal to use Oracle

This section provides information on how to manually transfer data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- Prerequisites
  - Creating users
  - Creating databases
  - Modifying database properties
  - Setting up databases
  - Creating JCR table spaces

Tips:
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

Steps for transferring data to another supported database

1. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.
2. Enter the following commands to validate configuration properties.

   ```bash
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ```
3. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.
4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td>./stopServer.sh server1 -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>
5. Transfer the database. **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

A. Change to the directory `wp_profile_root/ConfigEngine`.

B. Enter the following command: `./ConfigEngine.sh database-transfer -DWasPassword=password`  
   **Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command: `./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password`  

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example: `SQL> execute dbms_stats.gather_schema_stats(ownname=> 'jcr', cascade=> TRUE);`

7. Change to the directory `wp_profile_root/bin`.

8. Enter the following command to start the WebSphere Portal server: `./startServer.sh WebSphere_Portal`

---

**Parent topic:** Configuring WebSphere Portal to use Oracle  
**Previous topic:** Assigning custom table spaces
Configuring WebSphere Portal to use Oracle RAC

View information on installing and setting up Oracle RAC to work with WebSphere Portal.

1. Installing Oracle RAC
   View information on how to install Oracle RAC for use with WebSphere Portal.

2. Create users
   View the steps to set up users for Oracle RAC to work with WebSphere Portal.

3. Creating databases
   View some important considerations before setting up Oracle databases to work with WebSphere Portal.

4. Modifying database properties
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

6. Creating JCR table spaces
   This topic provides manual instructions for creating JCR table spaces for Oracle.

7. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

8. Configuring WebSphere Portal to use Oracle RAC
   This section provides information on how to manually transfer data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configure WebSphere Portal to use a remote database
Installing Oracle RAC

View information on how to install Oracle RAC for use with WebSphere Portal.

Before you begin:

- You should have completed reviewing the Planning for Oracle RAC section.
- Ensure the Oracle CRS (Cluster Ready Service) and Oracle RAC databases have been installed and configured on the primary and secondary nodes.
- Run the following commands on both RAC nodes to start global services daemon (GSD), oracle listeners, and agents.

  $ gsdctl start
  $ lsnrctl start
  $ agentctl start

You can use Oracle RAC as the database software for WebSphere Portal. You must install Oracle RAC separately and before performing WebSphere Portal database transfer.

Refer to Oracle product documentation for installation instructions.

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC

**Next topic:** Create users
Create users

View the steps to set up users for Oracle RAC to work with WebSphere Portal.

Before you begin:

- You should have completed installing Oracle RAC.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

**Prerequisites**

- Installing Oracle RAC

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Create the users in the WebSphere Portal databases
   A. Log in to the database in which you want to create the new users.
   B. Create the users that you need in the appropriate databases by entering the following command in the SQL*Plus tool, where `user_tablespace` is the default tablespace that is identified by the database administrator to store user objects and `temp_tablespace` is identified to store temporary objects: Tip: Balance the storage of user objects among tablespaces to prevent running out of space with overuse of `user_tablespace`. Also consider increasing the size of `user_tablespace` when handling a high volume of users.

   ```sql
   SQL> create user username identified by password
       default tablespace user_tablespace
       temporary tablespace temp_tablespace;
   ```

2. Log in by entering the command `$ sqlplus` in SQL*Plus:

3. Enter the command `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example: `system/manager@wpsdb` will log the administrative user system with a password of manager into the wpsdb database.

4. Create the WebSphere Portal user `dbdomain.DbUser`, where `dbdomain` is replaced by release, community, customization, or JCR. All names must be in lower case. The following code uses the default user tablespace and the default temporary tablespace. Explicitly specify these tablespaces if you do not want to use the defaults.

   ```sql
   SQL> create user releaseusr identified by password;
   SQL> create user communityusr identified by password;
   SQL> create user customizationusr identified by password;
   SQL> create user jcr identified by password;
   ```

5. Connect to the content database by entering the command `SQL> connect`.

6. Enter the following, where `username` is an existing administrative user in the database. `user-name: username/password@dbname` For example, `system/manager@jcrdb` will log the administrative user system with a password of manager into the Oracle RAC database.

7. Create the Oracle RAC users by entering the following: When creating the `jcr` user, grant all necessary privileges. If you do not grant privileges, you will receive the error `ICMADMIN lacks CREATE SESSION Privilege logon denied`. 

 Portal, Version 6.1.5
 Operating systems: AIX, HP-UX, i5/OS, Linux, Solaris, Windows

1180
when you try to connect with the `jcr` user:

```
SQL> create user jcr identified by password default tablespace users temporary tablespace temp;
```

8. Connect to the Feedback database:
   A. Enter the following command: `SQL> connect` 
   B. Enter `user-name: username/password@dbname` where `username` is an existing administrative user in the database. For example, `system/manager@fdbkdb` will log the administrative user system with a password of manager into the `fdbkdb` database.

9. Create the Feedback user:

```
SQL> create user feedback identified by password default tablespace users temporary tablespace temp;
```

10. Connect to the LikeMinds database:
```
SQL> connect
```

11. Enter `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example, `system/manager@lmdb` will log the administrative user system with a password of manager into the `lmdb` database.

12. Create the LikeMinds user:
```
SQL> create user lmdbusr identified by password default tablespace users temporary tablespace temp;
```

13. Log out of the command line tool using the command `SQL> exit`.

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC  
**Previous topic:** Installing Oracle RAC  
**Next topic:** Creating databases
Creating databases

View some important considerations before setting up Oracle databases to work with WebSphere Portal.

For information about creating databases, refer to the Oracle product documentation. For information on the recommended database architecture and the databases you will need to create, see the Planning for Oracle topic. Be sure that all databases to be used with WebSphere Portal are created as UNICODE character set databases.

If you are using remote Oracle 9 or 10 databases, you must also copy the ojdbc14.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM® WebSphere® Application Server Version 7.0, you must also copy the ojdbc6.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM WebSphere Application Server Version 6.1, you must also copy the ojdbc5.jar file from the remote Oracle server to the WebSphere Portal machine. The typical location is the oracle_home/sqldeveloper/jdbc/lib directory. Record the copy location on your local machine for future reference.

When creating Oracle databases for use with WebSphere Portal, you should consider the following information:

- The Oracle databases must be created manually before configuring WebSphere Portal.
- All databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle server will be remote or local to the WebSphere Portal installation.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle database in order for WebSphere Portal to communicate with the Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes:

  - `db_block_size = 8192` bytes
  - `db_cache_size = 307,200` bytes
  - `db_files = 1024` files
  - `log_buffer = 65536` bytes
  - `open_cursors = 1500` cursors
  - `pga_aggregate_target = 204,800` bytes
  - `pre_page_sga = true`
  - `processes = 300` processes
  - `shared_pool_size = 204,800` bytes

  **Note:** If you are using IBM Java Content Repository, the `open_cursors` value may need to be increased based on the table count in the Java Content Repository schema.

- Raise the number of parallel servers as appropriate. For example, if you have more than 875 parallel servers, you should set the `parallel_max_servers` to 1200.

- The Oracle parameter CURSOR_SHARING allows similar SQL Statements to be shared when possible, which prevents parsing and establishing a new execution plan. The execution plan is used by Oracle to gather the data needed to satisfy a request. There are two options for CURSOR_SHARING, which are as follows:

  - **FORCE**
    - When you select this option, Oracle uses the same execution plan for all SQLs that are similar in value even if the values are different. When you use this option, the execution plan may not provide optimum performance. For example, similar SQLs with different values may behave differently when executed running the same plan.

  - **EXACT**
- When you select this option, Oracle only shares the same execution plan for SQLs that are identical and use the same values. This option removes the risk of a SQL statement being executed when optimum performance conditions do not exist.

WebSphere Portal supports both options. Regardless of the option selected, portlet applications should not be affected. Contact your database administrator for further assistance on these options.

- **Prerequisites**
  - Installing Oracle RAC
  - Create users

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC

**Previous topic:** Create users

**Next topic:** Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- Prerequisites
  - Installing Oracle RAC
  - Create users
  - Creating databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

Tip: For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Working with properties files:
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:

  - release.DbName=release
  - jcr.DbName=jcrdb
  - feedback.DbName=fdbkdb
  - likeminds.DbName=lmdb
  - community.DbName=commdb
  - customization.DbName=custdb

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds

- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
- `dbdomainDbType`
- `dbdomainDbName`
- `dbdomainDbUrl`
- `dbdomainDbSchema`

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If `DbUser`, `DbUrl`, and `DbPassword` are not the same across domains, the value for `DataSourceName` must differ from the `DataSourceName` of the other domains. In other words, this value must be unique for the database domain.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

1. Locate the following files and create a backup copy of each before changing any values:
   - `wp_profile_root/ConfigEngine/properties/wkplc.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_comp.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties`

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In `wkplc_comp.properties`, most properties are repeated for each domain.

2. Use a text editor to open the properties file `wkplc_comp.properties` and modify the values to correspond to your environment.
   A. For `dbdomainDbType`, type `oracle`.
   B. For `dbdomainDbName`, type the name of the WebSphere Portal domain database. Note: This value is also the database element in the `dbdomainDbUrl` property.
   C. For `dbdomainDbSchema`, type the schema name of the database domain. Note: Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The `dbdomainDbName` should be the same value used for the `dbdomainDbSchema`.

   **Restriction**: The value for `dbdomainDbSchema` must equal the value for `dbdomainDbUser`.

   D. For `dbdomainDataSourceName`, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - `releaseDS`
      - `communityDS`
      - `customizationDS`
      - `jcrDS`
      - `lmdbDS`
      - `feedback`

   E. For `dbdomainDbUrl`, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. Note:
      - The database element of this value should match the value of `DbName`.

      For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following: `jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME`. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.
For dbdomain.DbUser, type the user ID for the database administrator. **Restriction:** The value for dbdomain .DbUser must equal the value for dbdomain.DbSchema.

G. For dbdomain.DbPassword, type the password for the database administrator.

H. For dbdomain.DBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.

I. For dbdomain.DBA.DbPassword, type the database administrator password for privileged access operations during creation of the database.

J. For dbdomain.DbHome, type the root location for the database. **Note:** This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For source.domainDbType, type the database you are currently configured to use. The value for source.domainDbType is Derby by default.

B. For source.domainDbName, type the name of the database domain you are currently using.

C. For source.domainDbSchema, type current schema identifier for objects within the database for this domain.

D. For source.domainDataSourceName, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For source.domainDbUrl, type the url currently used to access your database.

F. For source.domainDbUser, type the name of the user accessing this database.

G. For source.domainDbPassword, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file wkplc_dbtype.properties.

A. For oracleDbType, type the name of the class that SqlProcessor uses to import SQL files.

B. For oracleDbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

C. For oracleJdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file wkplc.properties.

A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC  
**Previous topic:** Creating databases  
**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

Before you begin, ensure that the following prerequisites are met:
- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

- Prerequisites
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties

1. On the database server, make sure that the folder `db_1` contains the subfolders `your_oracle_instance/data` and `your_oracle_instance/index`. If this folder hierarchy does not exist, create it manually before you run the `setup-database` task. The `setup-database` task requires these folders to create database users. If these folders do not exist, the `setup-database` task will fail.

2. Change to the directory `wp_profile_root/ConfigEngine`

3. To create the database users, type the following command:

   ```bash
   ./ConfigEngine.sh setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:

   ```sql
   '<WP_root>/base/wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql'.
   ```

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC

**Previous topic:** Modifying database properties

**Next topic:** Creating JCR table spaces

**Related tasks**

Manually creating users and granting privileges for Oracle
Creating JCR table spaces

This topic provides manual instructions for creating JCR table spaces for Oracle.

- **Prerequisites**
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties
  - Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.

2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:
   - `&jcrdb` is the name of the database you created to store user data.
   - `&dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.

   Ensure that the `.` is included in the variables when you substitute the values of your environment with these variables.

   **Important:** You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

   ```sql
   create tablespace ICMLFQ32 datafile '&dbpath./&jcrdb.data/ICMLFQ32_01.dbf' size 300M reuse autoextend on
   next 10M maxsize UNLIMITED extent management local autoallocate;
   
   create tablespace ICMLNF32 datafile '&dbpath./&jcrdb.data/ICMLNF32_01.dbf' size 25M reuse autoextend on
   next 10M maxsize UNLIMITED extent management local autoallocate;
   
   create tablespace ICMVFQ04 datafile '&dbpath./&jcrdb.data/ICMVFQ04_01.dbf' size 25M reuse autoextend on
   next 10M maxsize UNLIMITED extent management local autoallocate;
   
   create tablespace ICMSFQ04 datafile '&dbpath./&jcrdb.data/ICMSFQ04_01.dbf' size 150M reuse autoextend on
   next 10M maxsize UNLIMITED extent management local autoallocate;
   
   create tablespace ICMLSNDX datafile '&dbpath./&jcrdb.index/ICMLSNDX_01.dbf' size 10M reuse autoextend on
   next 10M maxsize UNLIMITED extent management local autoallocate;
   
   A. Set the `size`, `autoextend`, and `maxsize` values according to your environment. For example, you may want to change the `maxsize` to a set value rather than `UNLIMITED`.
   
   B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.
   
   C. Refer to the Oracle command reference for more information about using the `create tablespace` command.

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC  
**Previous topic:** Setting up databases  
**Next topic:** Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:

- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root`/PortalServer/config/tables/pages/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

**Prerequisites**

- Installing Oracle RAC
- Create users
- Creating databases
- Modifying database properties
- Setting up databases
- Creating JCR table spaces

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tables/pages/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`

   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - `release`
   - `community`
   - `customization`
   - `jcr`
   - `feedback`
   - `likeminds`

3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword `TABLESPACE` and a space. For example: `community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE`

   Repeat this step for each domain that you are transferring.
4. Save and close `dbdomain.space_mapping.properties`.
5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - Windows: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
- UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

Parent topic: Configuring WebSphere Portal to use Oracle RAC
Previous topic: Creating JCR table spaces
Next topic: Configuring WebSphere Portal to use Oracle RAC
Configuring WebSphere Portal to use Oracle RAC

This section provides information on how to manually transfer data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- Prerequisites
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties
  - Setting up databases
  - Creating JCR table spaces

Tips:
- If you are transferring from Oracle, the `open_cursors` setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

- The WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following: `jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME`. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.
- Manually specify an RAC datasource URL in the WebSphere Application Server console. The URL should look like the following:
  ```
  jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICENAME))
  ```
- When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

Steps for transferring data to another supported database
1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.
2. Enter the following commands to validate configuration properties:

   ```sh
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWsPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWsPassword=password
   ```

3. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.
4. Stop both WebSphere Application Server and the WebSphere Portal server:

   ```
   Option                           Description
   WebSphere Application Server     ./stopServer.sh server1 -username admin_userid -password admin_password
   WebSphere Portal                 ./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password
   ```

5. Transfer the database: Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.
   A. Change to the directory `wp_profile_root/ConfigEngine`.
   B. Enter the following command:

   ```sh
   ./ConfigEngine.sh database-transfer -DWsPassword=password
   ```

   **Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:

   ```sh
   ./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWsPassword=password
   ```

   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

   If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example:

   ```sql
   SQL> execute dbms_stats.gather_schema_stats(ownname=> 'jcr', cascade=> TRUE);
   ```

7. Specify the JDBC URL to connect to the cluster:
   A. Login to the WebSphere Application Server Administrator Console
   B. Navigate to Resources > JDBC Providers
   C. If there is a value in the Node field, remove it and click Apply.
   D. For each Oracle JDBC provider, repeat the following steps:
      1. Click the provider name.
      2. Select Data Sources.
      3. Click the name of the data source.
      4. Under the Oracle data source properties, modify the URL attribute according to Oracle RAC configuration:

         ```
         jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICENAME)))
         ```

      5. Save your changes.

8. Change to the directory `wp_profile_root/bin`.
9. Enter the following command to start the WebSphere Portal server:

   ```sh
   ./startServer.sh WebSphere_Portal
   ```

Parent topic: Configuring WebSphere Portal to use Oracle RAC
Previous topic: Assigning custom table spaces
Configuring WebSphere Portal to use SQL Server

View information on installing and setting up SQL Server to work with WebSphere Portal.

1. Installing SQL Server
   View the steps to install SQL Server for use with WebSphere Portal.

2. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

3. Optional: Assigning custom filegroups
   The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

4. Configuring WebSphere Portal to use SQL Server 2005
   This section provides information on how to manually transfer data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configure WebSphere Portal to use a remote database
Installing SQL Server

View the steps to install SQL Server for use with WebSphere Portal.

Before you begin this task, complete the following prerequisites:

- You should have completed reviewing the Planning for SQL Server topic.
- You must install SQL Server separately from WebSphere Portal.
- WebSphere Portal requires either the DataDirect Connect for JDBC 3.7 JDBC driver or the Microsoft SQL Server JDBC Driver.
- You can obtain the DataDirect Driver from DataDirect Technologies. The JDBC driver consists of two parts: the JDBC driver libraries and JTA libraries. The JDBC driver libraries must be put on the WebSphere Portal system, while the JTA libraries are intended for SQL Server.
- You can obtain the Microsoft SQL Server JDBC Driver from Microsoft. See the Microsoft SQL Server product documentation for installation details.

The following section provides instructions for installing SQL Server for use with IBM® WebSphere® Application Server and WebSphere Portal. These steps are the same for both the DataDirect and Microsoft drivers unless noted.

1. Install SQL Server and all required patches.
2. Select the authentication mode for this installation:
   - Windows Authentication Mode
   - Mixed Mode (Windows Authentication and SQL Server Authentication)
     Either authentication mode allows the user who is specified as the database administrator in the wkplc_comp.properties file to log in to the SQL server. Mixed Mode authentication allows either a Windows user or a SQL server user, or both, to log in to the SQL Server.
3. In the SQL Server Setup panel, **Components to Install**, select the following components, which are required services for WebSphere Portal:
   - **SQL Server Database Services**
     - **Integration Services** The option Integration Services, creates the directory DTS/Binn, where you will need to copy files when you install Microsoft SQL Server JDBC drivers and enable XA connections.
4. Complete the installation using SQL Server documentation as a guide.
5. Enable TCP/IP connectivity in the **SQL Server Configuration Manager**.
6. Install the JDBC driver using one of these methods:
   - Installing DataDirect Connect for JDBC drivers on UNIX
   - Installing DataDirect Connect for JDBC drivers on Windows
   - Installing Microsoft SQL Server JDBC drivers and enabling XA connections
7. Network the connections for MSDTC in the Windows Component Services as described in Microsoft Help and Support.
8. Start SQL Server.

**Installing DataDirect Connect for JDBC drivers on UNIX**
1. Purchase and download DataDirect Connect for JDBC and save file 360connectjdbc.jar in a temporary work directory.
2. To create the required files, run the following command from the directory that contains 360connectjdbc.jar:

`jar -xvf 360connectjdbc.jar`

3. Run `/Installer.sh` in the same directory.

4. When installation is complete, change directory to `/installation/lib`, where `installation` is the directory in which you installed the DataDirect Connect for JDBC drivers.

5. Run the following command to change the permissions on the JAR files in the `installation/lib` directory:

`chmod 777 *.jar`

6. Change the ownership and group of these files by running the following commands in the `installation/lib` directory:

```bash
chgrp system_grp *.jar
chown root *.jar
```

Where `system_grp` is the system group as labeled by your operating system.

**Installing Microsoft SQL Server JDBC drivers and enabling XA connections**

1. Download and install the Microsoft SQL Server JDBC driver.

2. Copy file `sqljdbc_xa.dll` from the `xa` subdirectory to the `C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqljdbc_xa.dll` directory of the SQL Server installation.

3. Start the database server.

4. Ensure that the Distributed Transaction Coordinator has been started. The status can be verified in the list of services in the Computer Management console.

5. Start the Microsoft SQL Server Management Studio and connect to the local database engine as the system administrator, `sa`.

6. Select **File > Open > File** and select `xa_install.sql` from the subdirectory of the downloaded and extracted JDBC driver.

7. Execute the script by selecting **Query > Execute**. **Note:** Any warnings that appear in the messages section of the application window that say that stored procedures cannot be found can be safely ignored.


   A. Open the Windows Registry Editor (regedit) and navigate to the element `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSDTC\XADLL`

   B. From the menu bar, select **Edit > New > String Value** to create a new parameter named `sqljdbc_xa.dll` in that element.

   C. Change the value of the new parameter to the location of the `sqljdbc_xa.dll` file copied in Step 2 above, for example: `C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqljdbc_xa.dll`

**Parent topic:** Configuring WebSphere Portal to use SQL Server

**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

Before you begin, ensure that the following prerequisites are met:
- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

**Prerequisites**
- Installing SQL Server

1. Change to the directory `wp_profile_root/ConfigEngine`
2. To create the databases, type the following command:
   ```bash
   ./ConfigEngine.sh create-database -DWasPassword=password
   ```
3. To create the database users, type the following command:
   ```bash
   ./ConfigEngine.sh setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal: `<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql`.

**Parent topic:** Configuring WebSphere Portal to use SQL Server  
**Previous topic:** Installing SQL Server  
**Next topic:** Assigning custom filegroups

**Related tasks**
Create users and databases for SQL Server 2005 on AIX and UNIX
Assigning custom filegroups

The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

Before you begin:
- The custom filegroups must exist prior to the execution of the database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root`/PortalServer/config/tablesomes/dbdomain.space_mapping.properties file.
- For details on creating filegroups refer to the documentation for the database.

- **Prerequisites**
  - Installing SQL Server
  - Setting up databases

If custom filegroups are assigned, each must be assigned explicitly. The default filegroups can be used to contain database objects; however the name of the default file group must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom filegroups:

1. Determine the names of your custom filegroups.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tablesomes that specifies the table space and index space for each property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.indexspace.indexspace`

   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds

3. Assign a filespace to each entry in the mapping file. The filegroup name must be prepended by the keyword `ON` and a space. For example: `community.COMP_INST.tablespace=ON COMM8KSPACE`

   Repeat this step for each domain that you are transferring.
4. Save and close `dbdomain.space_mapping.properties`.
5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - **Windows**: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
   - **UNIX**: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
   - **i5/OS**: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

**Parent topic:** Configuring WebSphere Portal to use SQL Server
Previous topic: Setting up databases
Next topic: Configuring WebSphere Portal to use SQL Server 2005
Configuring WebSphere Portal to use SQL Server 2005

This section provides information on how to manually transfer data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005.

As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

**Prerequisites**

- Installing SQL Server
- Setting up databases

**Tips:**

- If you are transferring from Oracle, the `open_cursors` setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

**Steps for transferring data to another supported database**

1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.
2. Enter the following commands to validate configuration properties.
   ```
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```
3. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.
4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td><code>./stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td><code>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

5. Transfer the database:

   **Note:** Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory `wp_profile_root/ConfigEngine`.
   B. Enter the following command:

   ```
   ./ConfigEngine.sh database-transfer -DWasPassword=password
   ```

   **Note:** To select specific database domains to transfer, modify the -DTransferDomainList specified in the command to
include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command: `./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password`

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. Change to the directory `wp_profile_root/bin`.
7. Enter the following command to start the WebSphere Portal server: `./startServer.sh WebSphere_Portal`
8. Update the SQL Server 2005 statistics for Portal, and JCR databases by opening SQL Server Management Studio, selecting **New Query**, and running the following query: `use db_name exec sp_updatestats @resample='resample';`

**Parent topic:** Configuring WebSphere Portal to use SQL Server

**Previous topic:** Assigning custom filegroups
Verifying databases

After you configure IBM® WebSphere® Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.

- **Prerequisites**
  - [Technotes for database connectivity issues](#)

You can verify the database connection using IBM WebSphere Application Server or by opening WebSphere Portal in a browser.

- To verify that the WebSphere Portal application server is running by using WebSphere Application Server, complete these steps:
  1. Open the WebSphere Application Server administrative console by entering the following address in a browser:
     
     ```
     http://hostname.example.com:10027/ibm/console
     ```
     where `hostname.example.com` is the fully qualified host name of the machine where WebSphere Portal is running and 10027 is the default transport port that is created by WebSphere Application Server.
  2. Log into the administrative console.
  3. Depending on your version of WebSphere Application Server, click the appropriate option:
    - For **WebSphere Application Server Version 6.1**: Click **Resources & JDBC Providers**.
    - For **WebSphere Application Server Version 7.0**: Click **Resources > JDBC > JDBC Providers**
  4. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
  5. Select the name of the data source that is defined in wkplc_comp.properties. The default data source is `wpdbDS`.
  6. Select the name of the JDBC provider that is specified in wkplc_dbtype.properties. The default JDBC provider is `wpdbJDBC_dbtype`, where `dbtype` is replaced by the value that matches your environment.
  7. Click **Test Connection** to verify the database connection. If configuration parameters have been changed, you might need to restart WebSphere Application Server for the test to complete.

- To verify that the WebSphere Portal application server is running by opening WebSphere Portal in a browser, enter the following URL in a supported browser:
  ```
  http://hostname.example.com:10040/wps/portal
  ```
  where `hostname.example.com` is the fully qualified host name of the machine where WebSphere Portal is running and 10040 is the default transport port that is created by WebSphere Application Server.

**Parent topic:** Configure WebSphere Portal to use a remote database
Configuring the primary node to communicate with the deployment manager on AIX

After installing IBM® WebSphere® Application Server Network Deployment and installing IBM WebSphere Portal on the primary node, you must configure the primary node and the deployment manager to communicate with each other.

- Prerequisites
  - Installing WebSphere Portal on AIX on the primary node
  - Configure WebSphere Portal to use a remote database

Perform the following steps to configure WebSphere Portal to communicate with the deployment manager:

1. Run the `./ConfigEngine.sh collect-files-for-dmgr -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory of the primary node, to create an archive or compressed file containing all the files which need to be copied to the Deployment Manager. **Note:** The archive or compressed file will be placed in the `wp_profile_root/filesForDmgr` directory and the file will be called `filesForDmgr.zip`.

2. Stop the deployment manager.

3. Expand the `filesForDmgr.zip` file into the installation root directory of the Deployment Manager; for example in the `/opt/IBM/WebSphere/AppServer` directory. **Note:** If the Deployment Manager profile was not created in the default `AppServer/profiles/Dmgr01` directory, then the `metadata_wkplc.xml` file, located in the `AppServer/profiles/Dmgr01/config/.repository/metadata_wkplc.xml` directory in the zip file, must be placed into the correct Deployment Manager profile directory.

4. If the Deployment Manager profile is running on the same application server where WebSphere Portal is installed, remove the `com.ibm.ws.portletcontainer.deploytask_6.1.5.jar` file from the `AppServer/plugins` directory.

5. Start the deployment manager.

**Parent topic:** Preparing the primary node on AIX  
**Previous topic:** Configure WebSphere Portal to use a remote database  
**Next topic:** Removing search collections on AIX

**Related tasks**

- Configuring Portal Search in a cluster on AIX
Removing search collections on AIX

If you plan to use search in a cluster, you must configure a remote search server; see Configuring Portal Search in a cluster for information. If you created any search collections, you will need to recreate them on the remote search server. If your search collection has data, export the collection before deleting it and then import the data on the remote server.

- **Prerequisites**
  - Installing WebSphere Portal on AIX on the primary node
  - Configure WebSphere Portal to use a remote database
  - Configuring the primary node to communicate with the deployment manager on AIX

Perform the following steps to delete all existing search collections from the primary node:

1. Log on to WebSphere Portal.
2. Navigate to Administration > Search Administration > Manage Search and then click Search Collections.
3. Click the Delete Collection icon for each search collection and then click OK until they are all deleted.
4. Restart the WebSphere_Portal server and then navigate back to the Search Collections page to verify that all search collections have been deleted.

**Parent topic:** Preparing the primary node on AIX

**Previous topic:** Configuring the primary node to communicate with the deployment manager on AIX
Choosing the type of cluster to create on AIX

If you installed a IBM® WebSphere® Application Server Network Deployment, create a static cluster to handle failover requests. If you installed a IBM WebSphere Virtual Enterprise, create a dynamic cluster to balance member workloads.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on AIX
  - Preparing your AIX operating system
  - Preparing the primary node on AIX

Choose one of the following options to create a cluster:

- **Creating a static cluster on AIX**
  After installing IBM WebSphere Portal on the primary node, configuring a remote database, and preparing the primary node to communicate with the Deployment Manager, you can create your static cluster to handle failover requests.

- **Creating a new dynamic cluster on AIX using WebSphere Virtual Enterprise**
  After installing and configuring IBM WebSphere Portal on your primary node and all additional nodes, you can create a new dynamic cluster. A dynamic cluster monitors performance and load information and is able to dynamically create and remove cluster members based on the workload.

**Parent topic:** Setting up a cluster on AIX  
**Previous topic:** Preparing the primary node on AIX  
**Next topic:** Preparing a remote Web server when portal is installed on AIX
Creating a static cluster on AIX

After installing IBM® WebSphere® Portal on the primary node, configuring a remote database, and preparing the primary node to communicate with the Deployment Manager, you can create your static cluster to handle failover requests.

Perform the following steps to create the cluster:

1. If the Deployment Manager is configured to use a stand-alone LDAP user registry, update the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory, on the primary node with the stand-alone LDAP user registry property values from the Deployment Manager. You can find these settings under the VMM Stand-alone LDAP configuration heading.

   **Important:** Ensure that you set `WasUserid` and `WasPassword` to the Deployment Manager user ID and password.

2. Stop the `server1` and `WebSphere_Portal` servers on the primary node and ensure that the following parameters are set correctly in the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory:

   **Note:** Although you can add these parameters directly to any task that you run while creating your cluster, you may want to add them to the properties file while creating your cluster and then remove them when you are finished to keep your environment secure.

   A. Set `WasSoapPort` to the port used to connect remotely to the deployment manager.
   B. Set `WasRemoteHostName` to the full host name of the server used to remotely connect to the deployment manager.
   C. Verify that `WasPassword` is set to your deployment manager password.
   D. Verify that `PortalAdminPwd` is set to your WebSphere Portal password.
   E. Verify that `ClusterName` is set.
   F. Verify that `PrimaryNode` is set to `true`.

3. Make a backup copy of the `wp_profile_root/config/cells/cell_name/wim/config/wimconfig.xml` and `wp_profile_root/config/cells/cell_name/wim/model/wimxmlextension.xml`, if available, files.

4. Run the `./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=dmgr_password` task, from the `wp_profile_root/ConfigEngine` directory of the primary node.

   **Note:** If you want to specify custom ports for the nodeagent, add the `DPortPropsFile=full to portsfile` parameter to the `cluster-node-config-pre-federation` task. You can use the ports files that are found on the Setup CD for `WebSphere_Portal` and `server1` as a guide.

   **Note:** You may receive a message about accepting an SSL signer certificate. Failure to accept the SSL signer certificate will cause the script to fail. Alternatively, the `com.ibm.ssl.enableSignerExchangePrompt` flag can be enabled in the `ssl.client.props` file for "DefaultSSLSettings" in order to allow acceptance of the signer during the connection attempt.

   **Warning:** If the `cluster-node-config-pre-federation` fails for any reason, you must perform the following steps before rerunning the task:

   A. Remove the node if the `AddNode` task succeeded.
   B. Log on to the deployment manager and perform the following steps if the items exist:
      1. Remove all enterprise applications.
      2. Remove the `WebSphere_Portal` server definition.
      3. Remove the WebSphere Portal JDBC Provider.

   **Note:** If you are migrating from a version prior to Version 6.1.0 that contains Lookaside data, read the following technote: Migrating with Lookaside Data.
5. Optional: If you configured a database user registry or a property extension (lookaside) database, run the following task to set up access to the database drivers:

**Note:** You will also need to run this task if you migrated the primary node from a release prior to Version 6.1.0, even if you did not configure a database user registry or a property extension (lookaside) database.

A. Set the property value for `federated.db.DbType` if using a database user registry or set the property value for `la.DbType` if using a property extension database in the `wkplc.properties` file.

**Note:** If you migrated the primary node from a version prior to Version 6.1.0 and you are not using a database user registry or a property extension database, set the property value for `federated.db DbType` to the same value that is in the `wkplc.properties` file on the primary node.

B. Run the `./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWaspPassword=dmgr_password -DDDBDomain=la|federated db -DVmmNodeName=node_name -Db_type.NodeDbLibrary=local full path of the database jars` task from the `wp_profile_root/ConfigEngine` directory to create the variable used to access the VMM database jars.

**Note:** `VmmNodeName` is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The `db_type` in `db_type.NodeDbLibrary` should be set to the type of database you are using, for example `db2`.

6. Run the `./ConfigEngine.sh cluster-node-config-post-federation -DWaspPassword=dmgr_password` task.

7. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell's user registry. Run the `./ConfigEngine.sh wp-change-portal-admin-user -DWaspPassword=password -DNewAdminId=newadminid -DNewAdminPw=newpassword -DNewAdminGroupId=newadmingroupid` task, from the `wp_profile_root/ConfigEngine` directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

**Attention:** The password value for `-DWaspPassword` is the Deployment Manager administrative password.

**Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

**Fast path:** If the value for `newAdminGroupId` contains a space; for example `Software Group`, open the `wkplc.properties` file and add the values for `newAdminId`, `newAdminPw`, and `newAdminGroupId`. Save your changes and run the `ConfigEngine.bat wp-change-portal-admin-user -DWaspPassword=dmgr_password` task.

**Important:** If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user` task until after the `cluster-node-config-cluster-setup` (static cluster) or `cluster-node-config-dynamic-cluster-setup` (dynamic cluster) task completes. After running the `wp-change-portal-admin-user` task, start or restart the WebSphere_Portal server to use the updated administrator user ID.

**Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the `wp-change-portal-admin-user` task. `-DNewAdminPw` is an optional parameter to update the Administrative password in the `wkplc.properties` file if required.

8. Run the `./ConfigEngine.sh cluster-node-config-cluster-setup -DWaspPassword=dmgr_password` task.

9. Configure the cluster to use an external Web server to take advantage of features such as workload management.

Choose one of the following options:

- Configuring a Web server and an application server on separate machines (remote)
- Configuring a Web server and an application server profile on the same machine
- Configuring a Web server and a deployment manager profile on the same machine

**Note:** Start with the step about launching the Plug-ins installation wizard.

10. Perform the following steps to access the Web Content Management content through an external Web server:

A. Log on to the deployment manager administrative console.
B. Select Environment > WebSphere Variables.

C. From the Scope drop-down menu, select the Node=nodename, Server=servername option to narrow the scope of the listed variables, where Node=nodename is the node that contains the application server.

D. Update the WCM_HOST variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.

E. Update the WCM_PORT variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.

F. Perform the following steps to synchronize the node with the deployment manager:
   1. Select System Administration > Nodes.
   2. Select the node that you want to synchronize from the list.
   3. Click Full Resynchronize.

G. Log off of the deployment manager administrative console.

11. Run the following tasks to propagate your changes: **Note:** WebSphere_Portal_nodename is the name of the node's WebSphere Portal server; but if you customized the server name, the default name changes. If you are uncertain of the server name, run the serverstatus -all task to get a list of the server names and their status.

A. ./stopManager.sh -username admin_userid -password admin_password, from the dmgr_profile_root/bin directory

B. ./stopNode.sh -username admin_userid -password admin_password, from the wp_profile_root/bin directory

C. ./stopServer.sh WebSphere_Portal_nodename -username admin_userid -password admin_password

D. ./startManager.sh, from the dmgr_profile_root/bin directory

E. ./startNode.sh, from the wp_profile_root/bin directory

F. ./startServer.sh WebSphere_Portal_nodename

12. Change the task list settings to point to the cluster instead of a single server if using Portal business process integration. See one of the following sections in the Information Center for information:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross cell</td>
<td>Adding a BPI-enabled portal server to a managed cell in a cross-cell setup</td>
</tr>
<tr>
<td>Single cell</td>
<td>Adding a BPI-enabled portal server to a managed cell in a single-cell setup</td>
</tr>
</tbody>
</table>

**Parent topic:** Choosing the type of cluster to create on AIX

**Related tasks**

- Adding a BPI-enabled portal server to a managed cell in a cross-cell setup
- Adding a BPI-enabled portal server to a managed cell in a single-cell setup
Creating a new dynamic cluster on AIX using WebSphere Virtual Enterprise

After installing and configuring IBM® WebSphere® Portal on your primary node and all additional nodes, you can create a new dynamic cluster. A dynamic cluster monitors performance and load information and is able to dynamically create and remove cluster members based on the workload.

Before creating your dynamic cluster, install and configure the Deployment Manager and perform all the tasks under “Preparing the primary node.” On the Deployment Manager system, install WebSphere Virtual Enterprise and augment the deployment manager profile. On all WebSphere Portal nodes, install WebSphere Virtual Enterprise and augment the wp_profile profile. See Installing and configuring the product for information about installing WebSphere Virtual Enterprise. See Augmenting profiles for information about augmenting the Deployment Manager and wp_profile profiles.

- Prerequisites
  - PK97393 for all WebSphere Application Server Versions prior to 6.1.0.5
  - Installing and configuring the product:
    - Augmenting profiles

Perform the following steps to create a dynamic cluster:

1. Perform the following steps for on the primary node of the dynamic cluster:
   A. Prepare the node for the dynamic cluster; perform all tasks under Preparing the primary node.
   B. If the Deployment Manager is configured to use a stand-alone LDAP user registry, update the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory, on the primary node with the stand-alone LDAP user registry property values from the Deployment Manager. You can find these settings under the VMM Stand-alone LDAP configuration heading.
   C. Stop the server1 and WebSphere_Portal servers on the primary node and ensure that the following parameters are set correctly in the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory:
      1. Set WasSoapPort to the port used to connect remotely to the deployment manager.
      2. Set WasRemoteHostName to the full host name of the server used to remotely connect to the deployment manager.
      3. Verify that WasPassword is set to your deployment manager password.
      4. Verify that PortalAdminPwd is set to your WebSphere Portal password.
      5. Verify that ClusterName is set.
      6. Verify that PrimaryNode is set to true.
   D. Run the ./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=dmgr_password task, from the wp_profile_root/ConfigEngine directory of the primary node.

   Warning: If the cluster-node-config-pre-federation fails for any reason, you must perform the following steps before rerunning the task:
   1. Remove the node if the AddNode task succeeded.
   2. Log on to the deployment manager and perform the following steps if the items exist:
      A. Remove all enterprise applications.
      B. Remove the WebSphere_Portal server definition.
      C. Remove the WebSphere Portal JDBC Provider.

   Note: If you are migrating from a version prior to Version 6.1.0 that contains Lookaside data, read the following
E. Run the following task. Include each node name as a comma separated list in the command:

1. Set the property value for `federated.db.DbType` if using a database user registry or set the property value for `la.DbType` if using a property extension database in the `wkplc.properties` file. **Note:** If you migrated the primary node from a version prior to Version 6.1.0 and you are not using a database user registry or a property extension database, set the property value for `federated.db.DbType` to the same value that is in the `wkplc.properties` file on the primary node.

2. Run the
```
./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=dmgr_password -DDBDomain=la/federated.db -DVmmNodeName=node_name -Ddb_type.NodeDbLibrary=local full path of the database jars
```
task from the `wp_profile_root/ConfigEngine` directory to create the variable used to access the VMM database jars. **Note:** `VmmNodeName` is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The `db_type` in `db_type.NodeDbLibrary` should be set to the type of database you are using, for example `db2`.

F. Run the
```
./ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=dmgr_password
```
task.

G. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell's user registry. Run the
```
./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid
```
task, from the `wp_profile_root/ConfigEngine` directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

**Important:** If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user` task until after the `cluster-node-config-dynamic-cluster-setup` task completes. After running the `wp-change-portal-admin-user` task, start or restart the WebSphere_Portal server to use the updated administrator user ID.

**Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the `wp-change-portal-admin-user` task. `-DnewAdminPw` is an optional parameter to update the Administrative password in the `wkplc.properties` file if required.

2. Log on to the deployment manager administrative console.

3. Perform the following steps to create a node group:
   A. Click **System administration > Node groups**.
   B. Click **New**.
   C. Type the node group **Name**.
   D. Optional: Type any information about the node group in the **Description** text box.
   E. Click **OK**.
   F. Click the **Save** link to save your changes to the master configuration.

4. Perform the following steps to add members to the node group:
   A. Click **System administration > Node groups**.
   B. Click on the name of the node group that you want to add members to.
   C. Click **Node group members** under Additional Properties.
   D. Click **Add**.
   E. Select the primary node and then click **Add**.
   F. Click the **Save** link to save your changes to the master configuration.

5. Perform the following steps to create a dynamic cluster in the node group:
   A. Click **Servers > Dynamic clusters**.
   B. Click **New**.
C. Select WebSphere Application Server from the Server Type pull-down menu and then click Next.
D. Type the cluster name in the Dynamic cluster name text box and then click Next. Type the same value that you provided for the ClusterName parameter in the wkplc.properties file of your primary node.
E. Remove all default membership policies and then click Subexpression builder.
F. Enter the following information in the Subexpression builder window:
   1. Select and from the Logical operator pull-down menu.
   2. Select Nodegroup from the Select operand pull-down menu.
   3. Select Equals (=) from the Operator pull-down menu.
   4. Type the nodegroup name you created in the previous step in the Value text box.
   5. Click Generate subexpression.
   6. Click Append.
G. Click Preview membership to verify that all nodes included in the nodegroup display and then click Next.
H. Click the Create the cluster member using an existing server as a template radio button and then select the WebSphere_Portal server for the primary node from the pull-down menu.
I. Click Next.
J. Specify the dynamic cluster properties and then click Next.
K. Review the summary page to verify your actions and then click Finish.
L. Click the Save link to save your changes to the master configuration.

6. Define or verify the following parameters in the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory:

   A. Set ClusterName to the name of the new dynamic cluster.
   B. Verify that CellName is set to the deployment manager cell.
   C. Verify that NodeName is set to the local WebSphere Portal node.
   D. Set ServerName to the server that will be used for the dynamic cluster member on this node.

       Note: Log on to the deployment manager administrative console and click Dynamic Clusters > PortalCluster > Dynamic cluster members to find the name of the server used for the dynamic cluster.
   E. Verify that PrimaryNode is set to true.

7. Run the ./ConfigEngine.sh cluster-node-config-dynamic-cluster-setup -DWasPassword=dmgr_password task from the wp_profile_root/ConfigEngine directory to create the dynamic cluster.

       Note: This task may display a warning message since the cluster already exists but it will proceed to create the new dynamic cluster; therefore, the warning can be ignored.

8. Perform the following steps to access the Web Content Management content through an external Web server:

   A. Log on to the deployment manager administrative console.
   B. Select Environment > WebSphere Variables.
   C. From the Scope drop-down menu, select the Node=nodename, Server=servername option to narrow the scope of the listed variables, where Node=nodename is the node that contains the application server.
   D. Update the WCM_HOST variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.
   E. Update the WCM_PORT variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.
   F. Perform the following steps to synchronize the node with the deployment manager:
      1. Select System Administration > Nodes.
      2. Select the node that you want to synchronize from the list.
      3. Click Full Reresynchronize.
   G. Log off of the deployment manager administrative console.

9. Run the following tasks to propagate your changes:

       Note: WebSphere_Portal_nodename is the name of the node's WebSphere Portal server; but if you customized the server name, the default name changes. If you are uncertain of the
server name, run the `serverstatus -all` task to get a list of the server names and their status.

A. `./stopManager.sh -username admin_userid -password admin_password`, from the `dmgr_profile_root/bin` directory
B. `./stopNode.sh -username admin_userid -password admin_password`, from the `wp_profile_root/bin` directory
C. `./stopServer.sh WebSphere_Portal_nodename -username admin_userid -password admin_password`
D. `./startManager.sh`, from the `dmgr_profile_root/bin` directory
E. `./startNode.sh`, from the `wp_profile_root/bin` directory
F. `./startServer.sh WebSphere_Portal_nodename`

**Parent topic:** Choosing the type of cluster to create on AIX
Preparing a remote Web server when portal is installed on AIX

Install and configure the Web server plug-in, provided by IBM® WebSphere® Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

- Prerequisites
  - Preparing the WebSphere Application Server Deployment Manager on AIX
  - Preparing prerequisite and corequisite software on AIX
  - Preparing your AIX operating system
  - Preparing the primary node on AIX
  - Choosing the type of cluster to create on AIX

Perform the following steps to install and configure your Web server:

1. Install and configure the Web server; refer to the Web server documentation for information.
2. If using IBM Lotus® Domino®, edit the $NOTES.INI$ file on the Web server. Set the $HTTPEnableConnectorHeaders$ parameter to 1.
3. If using IBM HTTP Server or Apache Server, edit the $httpd.conf$ file on the Web server. Set the $AllowEncodedSlashes$ directive to on; the directive should be added at the root level as a global directive.
4. Stop the Web server.
5. Install and configure the Web server plug-in on the system where the Web server is located using the Plug-ins installation wizard provided with WebSphere Application Server. Refer to Selecting a Web server topology diagram and roadmap for information. Note: The WebSphere Application Server Web server instructions are for the default profile. If not using the default profile, refer to Configuring a Web server for a non-default profile.

   If using WebDAV: After successfully installing the Web server plug-in, locate and open your $plugin-cfg.xml$ file and set $AcceptAllContent$ to true.

   Important: Depending on how you use the Web server, you may need to adjust the $ServerIOTimeout$ value, which defines how long the plug-in should wait for a response from the application. The recommended minimum value is 60 but you may need to adjust this value higher if you are retrieving data from a database. To update this value, locate and open your $plugin-cfg.xml$ file and set $ServerIOTimeout$ to a value that is appropriate for your business needs. For additional information, see Common questions about the Web server plug-in.
6. If you are using Sun Web Server Version 7 update 8, read Technote 1448262 and perform the steps to resolve the HTTP 408/409 error.
7. Start the Web server.
Next topic: Configuring WebSphere Portal to use a user registry on AIX in a clustered environment
Configuring WebSphere Portal to use a user registry on AIX in a clustered environment

Configure user registry security on IBM® WebSphere® Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

Prior to configuring security, you should use the IBM WebSphere Application Server backupConfig task to create and store a backup of the IBM WebSphere Portal configuration; see backupConfig command for information.

- Prerequisites
  - Preparing prerequisite and corequisite software on AIX
  - Preparing your AIX operating system
  - Preparing the primary node on AIX
  - Choosing the type of cluster to create on AIX

Perform the following tasks to configure WebSphere Portal to use a user registry:

1. Preparing user registries on AIX
   Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

2. Choosing your user registry model on AIX in a clustered environment
   Choose between securing IBM WebSphere Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

Parent topic: Setting up a cluster on AIX

Previous topic: Preparing a remote Web server when portal is installed on AIX

Next topic: Preparing additional nodes on AIX
Preparing user registries on AIX

Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

Choose the appropriate LDAP server to install and setup:

- **Preparing a Domino Directory server**
  If you plan to use a Domino Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a SecureWay Security Server**
  If you plan to use a SecureWay Security Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Novell eDirectory**
  If you plan to use a Novell eDirectory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Sun Java System Directory Server**
  If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Tivoli Directory Server**
  If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

**Parent topic:** Configuring WebSphere Portal to use a user registry on AIX in a clustered environment

**Next topic:** Choosing your user registry model on AIX in a clustered environment
Preparing a Domino Directory server

If you plan to use a Domino Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Domino Directory:

1. Perform the following steps to install the Domino Directory:
   A. Go to Lotus Domino documentation.
   B. Select the appropriate version tab for your product.
   C. Click the Download/View online link for the Lotus Domino for multiple platforms Information Center.
   D. Click Domino Administrator Help > Installation > Installing and setting up Domino servers > Server installation > Installing Domino on Unix and perform this task.
   E. Click Domino Administrator Help > Installation > Installing and setting up Domino servers > The Domino server setup program and perform this task.

2. Perform the following steps as a guide to create the WebSphere Portal administrative user:
   A. Navigate to the People view of the Domino Directory and then click Add Person.
   B. Enter the following values in the New Person form to create the wpsbind user:
      - Last Name
        - wpsbind
      - User Name
        - wpsbind/Domain, where Domain is your Lotus Domino Internet domain
        - wpsbind
        Note: Make sure you enter two values in the User Name field, where the first value includes the Lotus Domino domain.
      - Short name/UserID
        - wpsbind
      - Internet password
        - wpsbind
   C. Click Save and Close to save the new person record for wpsbind and return to the People view.
   D. Click Add Person and enter the following values in the New Person form to create the wpsadmin user:
      - Last Name
        - wpsadmin, where wpsadmin in the user ID for the WebSphere Portal Administrator
      - User Name
        - wpsadmin/Domain, where Domain is your Lotus Domino Internet domain
        - wpsadmin
        Note: Make sure you enter two values in the User Name field, where the first value includes the Lotus Domino domain.
      - Short name/UserID
        - wpsadmin
      - Internet password
        - wpsadmin
   E. Click Save and Close to save the new person record for wpsadmin and return to the People view.
   F. Navigate to the Groups view and click Add Group.
   G. Enter the following values in the New Group form on the Basic tab:
      - Group name
- **wpsadmins**
  - **Note:** If you plan to configure WebSphere Portal for multiple user registries and your Lotus Domino LDAP will share a realm with another user registry, you must use the hierarchical naming convention for the group names, for example: wpsadmins/DominoDomain, to avoid unexpected results during WebSphere Portal runtime.

- **Group type**
  - Multi-purpose

- **Members**
  - wpsbind/DominoDomain
  - wpsadmin/DominoDomain
  - **Note:** You can add additional administrator users if required.

H. Click **Save and Close** to save the wpsadmins group with the wpsbind and wpsadmin users as members.

3. Perform the following steps to update the access control list for the Domino Directory:
   A. Open the names.nsf file in the Lotus Domino Administrator or Lotus Notes client.
   B. Click **File > Application > Access Control** from the main menu to open the access control list for the file.
   C. In the **Access Control List > Basics** panel, ensure that the wpsadmins group has either Author or Editor access.
   D. Add the following Role Types to the wpsadmins group:
      - GroupCreator
      - GroupModifier
      - UserCreator
      - UserModifier
   E. Click **OK**.

*Parent topic:* Preparing user registries on AIX
Preparing a SecureWay Security Server

If you plan to use a SecureWay Security Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare SecureWay Security Server:

1. Install SecureWay Security Server; refer to the IBM SecureWay™ Security Server for z/OS® and OS/390® for information.

2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every dc=yourco,dc=com with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.
   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on AIX
Preparing a Novell eDirectory

If you plan to use a Novell eDirectory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Novell eDirectory:

1. Install Novell eDirectory; refer to eDirectory for information.
2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every dc=yourco,dc=com with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.
   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on AIX
Preparing a Sun Java System Directory Server

If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Sun Java System Directory Server:

1. Install Sun Java System Directory Server; refer to Oracle software for information.

2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: `dc=yourcompany,dc=com`.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every `dc=yourco,dc=com` with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.
   G. Save your changes.

3. Follow the instructions provided with your directory server to import the LDIF file.
4. Stop and restart the LDAP server.

Parent topic:Preparing user registries on AIX
Preparing a Tivoli Directory Server

If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Tivoli Directory Server:

   
   **Note:** Due to a restriction in Tivoli Directory Server, users or groups must not contain a Turkish uppercase dotted I or lowercase dotted i in the DN as this will prevent correct retrieval of that user or group.

2. Perform the following steps to create the WebSphere Portal administrative user:
   
   - Optional: Perform the following steps to create a new directory suffix:
     
     1. Click the **Server Administration** folder, located in the left-hand navigation of the directory server console.
     2. Click the **Manage Server Properties** folder under the Server Administration folder and then select **Suffixes** on the main page.
     3. Type the **Base DN** name for the suffix; for example: `dc=yourcompany,dc=com`.
     4. Click **Add**.
     5. Click **OK** to save your changes.
     6. Stop and restart the LDAP server.

   - Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
     
     - Use the `PortalUsers.ldif` file as a working example and adapted appropriately to work with your LDAP server.
     - Use the `ContentUsers.ldif` file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.

   - Replace every `dc=yourco,dc=com` with your suffix.

   - Replace any prefixes and suffixes that are unique to your LDAP server.

   - You can specify user names other than `wpsadmin` and `wpsbind`. For security reasons, specify nontrivial passwords for these administrator accounts.

   - Optional: If using IBM Tivoli® Access Manager Version 5.1, set the **objectclasses** to `accessGroup`. If using Tivoli Access Manager Version 6, set the **objectclasses** to `groupOfNames`.

   - Save your changes.

   - Follow the instructions provided with your directory server to import the LDIF file.

   - Stop and restart the LDAP server.

Parent topic: Preparing user registries on AIX
Choosing your user registry model on AIX in a clustered environment

Choose between securing IBM® WebSphere® Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

Choose one of the following user registry models:

- **Configuring a stand-alone LDAP user registry on AIX in a clustered environment**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Configuring the default federated repository on AIX in a clustered environment**
  Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

- **Adapting the attribute configuration**
  After installing IBM WebSphere Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

**Parent topic:** Configuring WebSphere Portal to use a user registry on AIX in a clustered environment

**Previous topic:** Preparing user registries on AIX
Configuring a stand-alone LDAP user registry on AIX in a clustered environment

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Configuring a stand-alone LDAP user registry on AIX in a clustered environment**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry to store all user account information for authorization.

- **Configuring a stand-alone LDAP user registry over SSL on AIX in a clustered environment**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

**Parent topic:** Choosing your user registry model on AIX in a clustered environment
Configuring a stand-alone LDAP user registry on AIX in a clustered environment

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry to store all user account information for authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps. If you need to rerun the wp-modify-ldap-security task to change the LDAP repositories or because the task failed, you must choose a new name for the realm using the standalone.ldap.realm parameter or you can set ignoreDuplicateIDs=true in the wkplc.properties file, before rerunning the task.

Perform the following steps to configure a standalone LDAP user registry:

Note: Use the wp_security_xxx.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_security_xxx.properties helper file.

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

2. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM Stand-alone LDAP configuration heading:

   - standalone.ldap.id
   - standalone.ldap.host
   - standalone.ldap.port
   - standalone.ldap.blindDN
   - standalone.ldap.bindPassword
   - standalone.ldap.ldapServerType
   - standalone.ldap.userIdMap
   - standalone.ldap.groupIdMap
   - standalone.ldap.groupMemberIdMap
   - standalone.ldap.userFilter
   - standalone.ldap.groupFilter
   - standalone.ldap.serverId
   - standalone.ldap.serverPassword
   - standalone.ldap.realm
   - standalone.ldap.primaryAdminId
   - standalone.ldap.primaryAdminPassword
   - standalone.ldap.primaryPortalAdminId
   - standalone.ldap.primaryPortalAdminPassword
   - standalone.ldap.primaryPortalAdminGroup
   - standalone.ldap.baseDN
3. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading:
   Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.ldap.et.group.objectClasses`
   - `standalone.ldap.et.group.objectClassesForCreate`
   - `standalone.ldap.et.group.searchBases`
   - `standalone.ldap.et.personaccount.objectClasses`
   - `standalone.ldap.et.personaccount.objectClassesForCreate`
   - `standalone.ldap.et.personaccount.searchBases`

4. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attributes heading:
   Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.ldap.gm.groupMemberName`
   - `standalone.ldap.gm.objectClass`
   - `standalone.ldap.gm.scope`
   - `standalone.ldap.gm.dummyMember`

5. Required: Enter a value for the following required relative distinguished name (RDN®) parameters in the `wkplc.properties` file under the Default parent, RDN attribute heading:
   Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.ldap.personAccountParent`
   - `standalone.ldap.groupParent`
   - `standalone.ldap.personAccountRdnProperties`
   - `standalone.ldap.groupRdnProperties`

6. Save your changes to the `wkplc.properties` file.

7. Optional: Enter the following Web content authors parameters in the `wkplc_comp.properties` file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product.
   Note: See the `wkplc_comp.properties` file for specific information about the required parameters and for advanced parameters.
   - `WcmContentAuthorsGroupId`
   - `WcmContentAuthorsGroupCN`

8. Run the `./ConfigEngine.sh validate-standalone-ldap -DWasPassword=(password)` task to validate your LDAP server settings. Attention: If you have not deleted the default file repository, `WasPassword` is the value entered during installation and not a value found in your LDAP user registry.
   Note: During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press y then Enter.

9. Run the `./ConfigEngine.sh wp-modify-ldap-security -DWasPassword=(password)` task, from the `wp_profile_root/ConfigEngine` directory, to set the stand-alone LDAP user registry.

10. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

11. Run the `./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=(password)` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured LDAP user registry. Important: When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

12. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the
LDAP directory.

**Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

A. Edit the `wp_profile_root`/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties file.

B. Add the following lines to the file:

   `uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN`

   Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `./ConfigEngine.sh action=express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root`/ConfigEngine directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

13. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See **Enabling LDAP security after cluster creation** for instructions.

**Parent topic:** Configuring a stand-alone LDAP user registry on AIX in a clustered environment

**Related tasks**

- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation
Configuring a stand-alone LDAP user registry over SSL on AIX in a clustered environment

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to configure a standalone LDAP user registry over SSL:

Note: Use the wp_security_xxx.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_security_xxx.properties helper file.

1. To specify the LDAP server’s SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:
   - **Server trust store**
     A. Add the certificate to the trust store:
        1. Log in to the WebSphere Application Server Administrative Console.
        2. Navigate to Security > SSL certificate and key management > SSL configurations.
        3. Click the appropriate SSL configuration from the list. For example,
           - Stand-alone environments: NodeDefaultSSLSettings
           - Clustered environments: CellDefaultSSLSettings
           **Clustered environments:** Ensure the setting for SSL configuration for outbound connection matches your SSL settings.
        4. Click Key stores and certificates.
        5. Click the appropriate trust store from the list; for example, NodeDefaultTrustStore.
        6. Click Signer certificates, click Add, and then enter the following information:
           - Type the Alias the key store uses for the signer certificate.
           - Type the File name where the signer certificate is located.
        7. Click OK and then click Save to save the changes to the master configuration.
   B. Retrieve the certificate from the port:
      1. Log in to the WebSphere Application Server Administrative Console.
      2. Navigate to Security > SSL certificate and key management > SSL configurations.
      3. Click the appropriate SSL configuration from the list. For example,
Clustered environments: CellDefaultSSLSettings

Clustered environments: Ensure the setting for SSL configuration for outbound connection matches your SSL settings.

4. Click Key stores and certificates.

5. Click the appropriate trust store from the list; for example, NodeDefaultTrustStore.

6. Click Signer certificates, click Retrieve from port, and then enter the following information:
   - Type the Host name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the SSL Port used when attempting to retrieve the signer certificate.
   - Type the Alias the key store uses for the signer certificate.

7. Click Retrieve signer information to retrieve the certificate from the port.

8. Click OK and then click Save to save the changes to the master configuration.

- Client trust storeNote: This task might report an error, but it does successfully update the trust store. You can ignore the error message.
  A. See Secure installation for client signer retrieval.
  B. Run the retrieveSigners task from the wp_profile_root/bin directory; see retrieveSigners command for information. In a deployed environment, you will need to run the retrieveSigners task, for any federated node, against the Deployment Manager. Note: This task might report an error, but it does successfully update the trust store. You can ignore the error message.

C. Update the trust store properties file:
   1. Use a text editor to open the ssl.client.props file, located in the wp_profile_root/properties directory.
   2. Change the com.ibm.ssl.trustStore parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter com.ibm.ssl.trustStore=${CONFIG_ROOT}\cells\wpsbvt\nodes\wpsbvt\trust.p12 to use the default trust store.
   3. Save your changes.

2. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

3. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM Stand-alone LDAP configuration heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.idap.id
   - standalone.idap.host
   - standalone.idap.port
   - standalone.idap.blndDN
   - standalone.idap.bindPassword
   - standalone.idap.idapServerType
   - standalone.idap.userldMap
   - standalone.idap.groupIdMap
   - standalone.idap.groupMemberIdMap
   - standalone.idap.userFilter
   - standalone.idap.groupFilter
   - standalone.idap.serverId
   - standalone.idap.serverPassword
   - standalone.idap.realm
   - standalone.idap.prlmaryAdminId
   - standalone.idap.prlmaryAdminPassword
   - standalone.idap.prlmaryPortalAdminId

1231
4. Required: Enter a value for the following required entity types parameters in the wkplc.properties file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.et.group.objectClasses
   - standalone.ldap.et.group.objectClassesForCreate
   - standalone.ldap.et.group.searchBases
   - standalone.ldap.et.personaccount.objectClasses
   - standalone.ldap.et.personaccount.objectClassesForCreate
   - standalone.ldap.et.personaccount.searchBases

5. Required: Enter a value for the following required group member parameters in the wkplc.properties file under the Group member attributes heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.gm.groupMemberName
   - standalone.ldap.gm.objectClass
   - standalone.ldap.gm.scope
   - standalone.ldap.gm.dummyMember

6. Required: Enter a value for the following required relative distinguished name (RDN®) parameters in the wkplc.properties file under the Default parent, RDN attribute heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.personAccountParent
   - standalone.ldap.groupParent
   - standalone.ldap.personAccountRdnProperties
   - standalone.ldap.groupRdnProperties

7. Enter a value for the following parameters to enable Secure Socket Layers (SSL): **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   Required parameters:
   - standalone.ldap.sslEnabled
   - standalone.ldap.sslConfiguration

   Optional parameters:
   - standalone.ldap.certificateMapMode
   - standalone.ldap.certificateFilter

8. Save your changes to the wkplc.properties file.

9. Run the `./ConfigEngine.sh validate-standalone-ldap -DWasPassword=password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, **WasPassword** is the value entered during installation and not a value found in your LDAP user registry.
   **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press `y` then **Enter**.

10. Run the `./ConfigEngine.sh wp-modify-ldap-security -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to set the stand-alone LDAP user registry.

11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: **Starting and stopping servers, deployment managers, and node agents.**

12. Run the `./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured...
LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring a stand-alone LDAP user registry on AIX in a clustered environment

**Related tasks**
- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation
Configuring the default federated repository on AIX in a clustered environment

Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

Perform the following tasks to add user registries to the default federated repository:

- **Configuring a federated LDAP user registry on AIX in a clustered environment**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Adding a database user registry on AIX in a clustered environment**
  Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

- **Adding realm support on AIX in a clustered environment**
  A realm is a group of users from one or more user registries that form a coherent group within IBM WebSphere Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

**Parent topic:** Choosing your user registry model on AIX in a clustered environment
Configuring a federated LDAP user registry on AIX in a clustered environment

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Adding an LDAP user registry on AIX in a clustered environment**
  Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time. If IBM Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=groupName` and the hierarchical format is `cn=groupName, o=root`. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as `wpsadmin`, this ID cannot exist in the LDAP you are adding.

- **Adding an LDAP user registry over SSL on AIX in a clustered environment**
  Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

**Parent topic:** Configuring the default federated repository on AIX in a clustered environment
Adding an LDAP user registry on AIX in a clustered environment

Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time. If IBM® Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=groupName` and the hierarchical format is `cn=groupName,o=root`. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as `wpsadmin`, this ID cannot exist in the LDAP you are adding.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

**Tip:** Perform these steps on the primary node only.

**Note:** Use the `wp_add_federated_xxx.properties` helper file, located in the `wp_profile_root` /ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_add_federated_xxx.properties` helper file.

1. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root`/ConfigEngine/properties directory.

2. Required: Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated LDAP Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - `federated.ldap.id`
   - `federated.ldap.host`
   - `federated.ldap.port`
   - `federated.ldap.bindDN`
   - `federated.ldap.bindPassword`
   - `federated.ldap.ldapServerType`
   - `federated.ldap.baseDN`

3. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - `federated.ldap.et.group.objectClasses`
   - `federated.ldap.et.group.objectClassesForCreate`
   - `federated.ldap.et.group.searchBases`
   - `federated.ldap.et.personaccount.objectClasses`
- `federated.ldap.et.personaccount.objectClassesForCreate`
- `federated.ldap.et.personaccount.searchBases`

4. **Required:** Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attribute heading:
   
   **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   
   - `federated.ldap.gm.groupMemberName`
   - `federated.ldap.gm.objectClass`
   - `federated.ldap.gm.scope`
   - `federated.ldap.gm.dummyMember`

5. Save your changes to the `wkplc.properties` file.

6. Run the `./ConfigEngine.sh validate-federated-ldap -DWasPassword=password` task to validate your LDAP server settings.
   
   **Attention:** If you have not deleted the default file repository, `WasPassword` is the value entered during installation and not a value found in your LDAP user registry.
   
   **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now?.
   
   Press **y** then **Enter**.

7. Run the `./ConfigEngine.sh wp-create-ldap -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to add an LDAP user registry to the default federated repository.
   
   **Note:** Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: "Starting and stopping servers, deployment managers, and node agents."

9. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:
   
   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
   
   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms:
      
      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
      
      - `id`
      - `baseDN`
      - `nameInRepository`
   
   C. Save your changes to the `wkplc.properties` file.
   
   D. Run the `./ConfigEngine.sh wp-create-base-entry -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to create a base entry in a repository.
   
   E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Run the `./ConfigEngine.sh wp-query-repository -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to list the names and types of configured repositories.

11. Run the `./ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

12. Perform the following steps to update the user registry where new users and groups are stored:
   
   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties`
and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

   - `personAccountParent`
   - `groupParent`
   - `personAccountRdnProperties`
   - `groupRdnProperties`

   The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:
   - `personAccountParent=dc=yourco,dc=com`
   - `groupParent=dc=yourco,dc=com`

C. Save your changes to the `wkplc.properties` file.

D. Run the `./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to delete the old attributes before adding the new attributes.

E. Stop and restart all necessary servers to propagate your changes.

13. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. Enter a value for `realmName` or leave blank to update the default realm.

C. Save your changes to the `wkplc.properties` file.

D. Run the `./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

E. Stop and restart all necessary servers to propagate your changes.

14. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

   **Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

A. Edit the `wp_profile_root/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties` file.

B. Add the following lines to the file:

   ```
   uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
   ```

   Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `./ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: **Table 1. Value for realm_name when running the Member Fixer task to update the member names**
15. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

16. Optional: **This step is required in a production environment.** Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

   **Important:**
   - Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
   - Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

   **Note:** If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

   A. Run the following task from the `wp_profile_root/ConfigEngine` directory:

   ```bash
   .ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword
   **Important:** You must provide the full distinguished name (DN) for the `newAdminId` parameter.
   ```

   B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

   C. Run the following task from the `wp_profile_root/ConfigEngine` directory:

   ```bash
   .ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid
   **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.
   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.
   ```

   D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

17. Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it. The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: Deleting the repository.

   If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See Enabling LDAP security after cluster creation for instructions.
Parent topic: Configuring a federated LDAP user registry on AIX in a clustered environment

Related tasks
Starting and stopping servers, deployment managers, and node agents
Enabling LDAP security after cluster creation

Related information
User IDs and passwords
Adding an LDAP user registry over SSL on AIX in a clustered environment

Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry over SSL to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

**Tip:** Perform these steps on the primary node only.

**Note:** Use the `wp_add_federated_xxx.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_add_federated_xxx.properties` helper file.

1. To specify the LDAP server’s SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:
   - **Server trust store**
     A. Add the certificate to the trust store:
        1. Log in to the WebSphere Application Server Administrative Console.
        2. Navigate to **Security > SSL certificate and key management > SSL configurations**.
        3. Click the appropriate SSL configuration from the list. For example,
           - **Stand-alone environments**: `NodeDefaultSSLSettings`
           - **Clustered environments**: `CellDefaultSSLSettings`
        4. **Clustered environments**: Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.
        5. Click **Key stores and certificates**.
        6. Click **Signer certificates**, click **Add**, and then enter the following information:
           - Type the **Alias** the key store uses for the signer certificate.
           - Type the **File name** where the signer certificate is located.
        7. Click **OK** and then click **Save** to save the changes to the master configuration.
     B. Retrieve the certificate from the port:
        1. Log in to the WebSphere Application Server Administrative Console.
2. Navigate to **Security > SSL certificate and key management > SSL configurations**.
3. Click the appropriate SSL configuration from the list. For example,
   - Stand-alone environments: **NodeDefaultSSLSettings**
   - Clustered environments: **CellDefaultSSLSettings**

   **Clustered environments:** Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.
4. Click **Key stores and certificates**.
5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.
6. Click **Signer certificates**. Click **Retrieve from port**, and then enter the following information:
   - Type the **Host** name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the **SSL Port** used when attempting to retrieve the signer certificate.
   - Type the **Alias** the key store uses for the signer certificate.
7. Click **Retrieve signer information** to retrieve the certificate from the port.
8. Click **OK** and then click **Save** to save the changes to the master configuration.

   **Client trust store**
   - **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

   **A.** See **Secure installation for client signer retrieval**.

   **B.** Run the **retrieveSigners** task from the **wp_profile_root/bin** directory; see **retrieveSigners command** for information. In a deployed environment, you will need to run the **retrieveSigners** task, for any federated node, against the **Deployment Manager**.

   **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

   **C.** Update the trust store properties file:
   1. Use a text editor to open the **ssl.client.props** file, located in the **wp_profile_root/properties** directory.
   2. Change the **com.ibm.ssl.trustStore** parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter

      com.ibm.ssl.trustStore=${CONFIG_ROOT}\cells\wpsbvt\nodes\wpsbvt\trust.p12 to use the default trust store.

   3. Save your changes.

2. Use a text editor to open the **wkplc.properties** file, located in the **wp_profile_root/ConfigEngine/properties** directory.
3. Required: Enter a value for the following required parameters in the **wkplc.properties** file under the VMM Federated LDAP Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - **federated.idap.id**
   - **federated.idap.host**
   - **federated.idap.port**
   - **federated.idap.bindDN**
   - **federated.idap.bindPassword**
   - **federated.idap.IdapServerType**
   - **federated.idap.baseDN**

4. Required: Enter a value for the following required entity types parameters in the **wkplc.properties** file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - **federated.idap.et.group.objectClasses**
   - **federated.idap.et.group.objectClassesForCreate**
   - **federated.idap.et.group.searchBases**
- federated.ldap.et.personaccount.objectClasses
- federated.ldap.et.personaccount.objectClassesForCreate
- federated.ldap.et.personaccount.searchBases

5. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attribute heading:
   **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

- federated.ldap.gm.groupMemberName
- federated.ldap.gm.objectClass
- federated.ldap.gm.scope
- federated.ldap.gm.dummyMember

6. Enter a value for the following parameters to enable Secure Socket Layers (SSL): **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   Required parameters:
   - federated.ldap.sslEnabled
   - federated.ldap.sslConfiguration

   Optional parameters:
   - federated.ldap.certificateMapMode
   - federated.ldap.certificateFilter

7. Save your changes to the `wkplc.properties` file.

8. Optional: Enter the following Web content authors parameters in the `wkplc_comp.properties` file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. **Note:** See the `wkplc_comp.properties` file for specific information about the required parameters and for advanced parameters.

- WcmContentAuthorsGroupId
- WcmContentAuthorsGroupCN

9. Run the `./ConfigEngine.sh validate-federated-ldap -DWasPassword=\password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, `WasPassword` is the value entered during installation and not a value found in your LDAP user registry.

   **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press `y` then `Enter`.

10. Run the `./ConfigEngine.sh wp-create-ldap -DWasPassword=\password` task, from the `wp_profile_root/ConfigEngine` directory, to add an LDAP user registry to the default federated repository. **Note:** Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

12. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms:

   **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
C. Save your changes to the \texttt{wkplc.properties} file.

D. Run the \texttt{./ConfigEngine.sh \ wp-create-base-entry -DWasPassword=password} task, from the \texttt{wp_profile_root/ConfigEngine} directory, to create a base entry in a repository.

E. Stop and restart all necessary servers to propagate your changes.

13. Optional: Run the \texttt{./ConfigEngine.sh \ wp-query-repository -DWasPassword=password} task, from the \texttt{wp_profile_root/ConfigEngine} directory, to list the names and types of configured repositories.

14. Run the \texttt{./ConfigEngine.sh \ wp-validate-federated-ldap-attribute-config -DWasPassword=password} task, from the \texttt{wp_profile_root/ConfigEngine} directory, to check that all defined attributes are available in the configured LDAP user registry.\textbf{Important:} When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

15. Perform the following steps to update the user registry where new users and groups are stored:

\textbf{Note:} If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

\textbf{Attention:} During installation, the default file repository creates a default value in the \texttt{personAccountRdnProperties} and \texttt{groupRdnProperties} parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

A. Use a text editor to open the \texttt{wkplc.properties} file, located in the \texttt{wp_profile_root/ConfigEngine/properties} directory.

B. Enter a value for the following required parameters in the \texttt{wkplc.properties} file under the VMM supported entity types configuration heading:

\textbf{Note:} See the properties file for specific information about the required parameters and for advanced parameters.

- \texttt{personAccountParent}
- \texttt{groupParent}
- \texttt{personAccountRdnProperties}
- \texttt{groupRdnProperties}

The parameters \texttt{groupParent} and \texttt{personAccountParent} must be set to the same value. For example:

- \texttt{personAccountParent=dc=yourco,dc=com}
- \texttt{groupParent=dc=yourco,dc=com}

C. Save your changes to the \texttt{wkplc.properties} file.

D. Run the \texttt{./ConfigEngine.sh \ wp-set-entitytypes -DWasPassword=password} task, from the \texttt{wp_profile_root/ConfigEngine} directory, to delete the old attributes before adding the new attributes.

E. Stop and restart all necessary servers to propagate your changes.

16. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

A. Use a text editor to open the \texttt{wkplc.properties} file, located in the \texttt{wp_profile_root/ConfigEngine/properties} directory.

B. Enter a value for \texttt{realmName} or leave blank to update the default realm.

C. Save your changes to the \texttt{wkplc.properties} file.

D. Run the \texttt{./ConfigEngine.sh \ wp-modify-realm-enable-dn-login -DWasPassword=password} task, located in the \texttt{wp_profile_root/ConfigEngine} directory, to enable the distinguished name login.\textbf{Note:} After running this task to enable the full distinguished name login, you can run the \texttt{./ConfigEngine.sh \ wp-modify-realm-disable-dn-login -DWasPassword=password} task to disable the feature.
17. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

Note: This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the configure-express task.

A. Edit the `wp_profile_root`/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties file.

B. Add the following lines to the file:

```
uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
```

Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `./ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory. Note: Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the wkplc.properties file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the wkplc.properties file. If the value for <code>federated.realm</code> is empty, use defaultWIMFileBasedRealm as the default value.</td>
</tr>
</tbody>
</table>

18. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

19. Optional: This step is required in a production environment. Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

   Important:
   - Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
   - Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

Note: If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

A. Run the following task from the `wp_profile_root/ConfigEngine` directory: `./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword` Important: You must provide the full distinguished name (DN) for the `newAdminId` parameter.
B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

C. Run the following task from the `wp_profile_root/ConfigEngine` directory:
```
./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid
```

**Important:** You must provide the full distinguished name (DN) for the newAdminId and newAdminGroupId parameters.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `--skip.ldap.validation=true` parameter to the task to skip the validation.

D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

20. Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it. The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: *Deleting the repository.*

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring a federated LDAP user registry on AIX in a clustered environment

**Related tasks**

- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation

**Related information**

- User IDs and passwords
Adding a database user registry on AIX in a clustered environment

Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps. If you have WebSphere Application Server Version 7.0.x installed, you must install APAR PM23090 and APAR PM24181 for WebSphere Portal prior to running this task. Clusters with WebSphere Application Server Version 6.1.x do not require these APARs.

Perform the following steps to add a database user registry to the default federated repository; you must repeat these steps for each additional database user registry that you plan to add:

**Tip:** Perform these steps on the primary node only.

**Note:** Use the `wp_add_DB.properties` helper file, located in the `/wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_add_DB.properties` helper file.

1. Set up a new database, including creating a new user with appropriate database privileges for accessing the database:
   - **Instructions for setting up databases:** Refer to the appropriate documentation for the type of database you want to set up.
   - **Consulting your database administrator:** The task of setting up a new database is typically performed by a database administrator. However, the following steps are provided for your reference in the event you are creating a stand-alone database for testing or demonstration purposes. Consult your database administrator before proceeding with the following steps if you plan to create a database for a production environment.

   **Table 1. Steps for creating a new database to use as a database user registry.**

<table>
<thead>
<tr>
<th>Database</th>
<th>Steps</th>
</tr>
</thead>
</table>

1247
### DB2
Perform the following steps to create a DB2 database:
- Install DB2.
- Enter the following database tuning commands:
  ```
  CREATE DB dbname using codeset UTF-8
territory us PAGESIZE 8192
  UPDATE DB CFG FOR dbname USING
  applheapsz 4096
  UPDATE DB CFG FOR dbname USING
  app_ctl_heap_sz 1024
  UPDATE DB CFG FOR dbname USING
  stmtheap 32768
  UPDATE DB CFG FOR dbname USING
  dbheap 2400
  UPDATE DB CFG FOR dbname USING
  locklist 1000
  UPDATE DB CFG FOR dbname USING
  logfilsz 4000
  UPDATE DB CFG FOR dbname USING
  logprimary 12
  UPDATE DB CFG FOR dbname USING
  logsecondary 20
  UPDATE DB CFG FOR dbname USING
  logbufsz 32
  UPDATE DB CFG FOR dbname USING
  avg_appls 5
  UPDATE DB CFG FOR dbname USING
  locktimeout 30
  UPDATE DB CFG FOR dbname using
  AUTO_MAINT off
  ```

### Oracle
Perform the following steps to create an Oracle database:
- Install Oracle using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- Configure the database in Dedicated Server Mode.
- Enter the recommended initial buffer pool sizes or set them according to your business needs:
  - `db_block_size = 8192`
  - `db_cache_size = 300M`
  - `db_files = 1024`
  - `log_buffer = 65536`
  - `open_cursors = 1500`
  - `pga_aggregate_target = 200M`
  - `pre_page_sga = true`
  - `processes = 300`
  - `shared_pool_size = 200M`

### SQL Server
Perform the following steps to create an SQL Server database:
- Install SQL Server.
- Set Collation to case-sensitive.

**Note:** Install SQL Server with the appropriate portal database collation so that your tempdb collation setting matches the collation you use for the property extension database. The tempdb collation is inherited from the master database, which you set when you install SQL Server.

2. Perform the following steps to define the **DbDriver** and **DbLibrary** parameter values:
   A. Navigate to the following directory: `wp_profile_root/ConfigEngine/properties`
   B. Locate and open `wkplc_dbtype.properties` with any text editor.
   C. Enter a value for the following parameters under the appropriate database type properties heading:
      - `db_type.DbDriver`
      - `db_type.DbLibrary`
   D. Save your changes.

3. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

4. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated Database Properties heading:
   **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
5. Change the value for the `com.ibm.SOAP.requestTimeout` parameter to 1000.

   A. Navigate to the following directory: `wp_profile_root/properties`
   B. Locate and open `soap.client.props` with any text editor.
   C. Locate the `com.ibm.SOAP.requestTimeout` parameter and change the value to 1000.
   D. Save and close `soap.client.props`.

6. Perform the following steps in a clustered environment:

   A. Run the `./ConfigEngine.sh wp-prep-vmm-db-secured-environment -DWasPassword=password` task from the `wp_profile_root/ConfigEngine` directory to create the local Deployment Manager WebSphere variable used to access the database jars. **Note:** The `db_type` in `db_type.DmgrDbLibrary` should be set to the type of database you are using, for example `db2`. The `local full path of the database jars on the Deployment Manager` should be one of the following options:

   - **DB2 Type 2 driver:** `db2java.zip`
   - **DB2 Type 4 driver:** `db2jcc.jar;db2jcc_license_cu.jar`
   - **DB2 for z/OS Type 2 driver:** `db2java.zip`
   - **DB2 for z/OS Type 4 driver:** `db2jcc.jar;db2jcc_license_cisuz.jar;db2jcc_javax.jar`
   - **Oracle:** `ojdbc14.jar`
   - **SQL Server JDBC driver provided by Microsoft:** `sqljdbc.jar`
   - **SQL Server JDBC driver provided by DataDirect:** `sqlserver.jar;base.jar;util.jar`

   B. Run the following task. Include each node name as a comma separated list in the command:

   **Running the task:** You do not have to run this task more than once. You can run this task from any node in the cluster.

   1. Set the property value for `federated.db.DbType` in the `wkpcl.properties` file if using a database user registry.
   2. Run the `./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=password -DbDomain=federated.db -Db_type.DmgrDbLibrary=local full path of the database jars on the Deployment Manager -DNodeDbName=dmgr_node_name` task from the `wp_profile_root/ConfigEngine` directory on each node to create the variable used to access the VMM database jars. **Note:** The `VmmNodeName` is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The `db_type` in `db_type.NodeDbLibrary` should be set to the type of database you are using, for example `db2`.
   C. Stop and restart all necessary servers to propagate your changes.

7. Run the `./ConfigEngine.sh wp-create-db -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to add a database user registry to the default federated repository. **Note:** Users who are not in an LDAP do...
not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

9. Perform the following steps to update the user registry where new users and groups are stored:

   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the **personAccountRdnProperties** and **groupRdnProperties** parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the **wkplc.properties** file, located in the **wp_profile_root/ConfigEngine/properties** directory.

   B. Enter a value for the following required parameters in the **wkplc.properties** file under the VMM supported entity types configuration heading:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - **personAccountParent**
      - **groupParent**
      - **personAccountRdnProperties**
      - **groupRdnProperties**

      The parameters **groupParent** and **personAccountParent** must be set to the same value. For example:

      - **personAccountParent=dc=yourco,dc=com**
      - **groupParent=dc=yourco,dc=com**

   C. Save your changes to the **wkplc.properties** file.

   D. Run the **./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password** task, from the **wp_profile_root/ConfigEngine** directory, to delete the old attributes before adding the new attributes.

   E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Run the **./ConfigEngine.sh wp-query-repository -DWasPassword=password** task, from the **wp_profile_root/ConfigEngine** directory, to list the names and types of configured repositories.

    If you created your clustered environment then performed the steps in this task, you must now run the **enable-jcr-security** task on the secondary node. See Enabling LDAP security after cluster creation for instructions.

**Parent topic:** Configuring the default federated repository on AIX in a clustered environment

**Related tasks**

Starting and stopping servers, deployment managers, and node agents

Enabling LDAP security after cluster creation
Adding realm support on AIX in a clustered environment

A realm is a group of users from one or more user registries that form a coherent group within IBM® WebSphere® Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

Before configuring realm support, you must add all LDAP user registries and/or database user registries, that you will use to create a single realm or multiple realms, to the federated repository. If you are going to create multiple realms, you must create all required base entries within your LDAP user registries and/or database user registries. All base entry names must be unique within the federated repository. In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add realm support to your user registry model:

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

2. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading:
   - realmName
   - securityUse
   - delimiter
   - addBaseEntry
   Note: See the properties file for specific information about the required parameters and for advanced parameters.

3. Save your changes to the wkplc.properties file.

4. Run the ./ConfigEngine.sh wp-create-realm -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add a new realm to the Virtual Member Manager configuration. Important: To create multiple realms, ensure that your federated repository contains the required unique base entries. Stop and restart the appropriate servers for your installation environment, and then update the wkplc.properties file with the base entry information and rerun the wp-create-realm task. Repeat these steps until all realms are created.

5. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

6. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading and then save your changes:
   - realmName
   - realm.personAccountParent
   - realm.groupParent
   - realm.orgContainerParent
   Note: See the properties file for specific information about the required parameters and for advanced parameters.

7. Run the ./ConfigEngine.sh wp-modify-realm-defaultparents -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to update the default parents per entity type and realm. Important: Stop and restart the appropriate servers for your installation environment before rerunning this task for any additional entity types.
8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

9. Optional: Perform the following steps to add additional base entries to the realm configuration; for example, if you had two additional base entries (base entry 1 and base entry 2) to add to the realm you just created, you would update the wkplc.properties file with the information from base entry 1 and then run this task. Then you would update the properties file with the information for base entry 2 and then run this task:
   A. Use a text editor to open the wkplc.properties file, located in the `wp_profile_root/ConfigEngine/properties` directory.
   B. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading:
      - `realmName`
      - `addBaseEntry`
   C. Save your changes to the wkplc.properties file.
   D. Run the `./ConfigEngine.sh wp-add-realm-baseentry -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to add additional LDAP base entries to the realm configuration.
   E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID; this step is required if you change the default realm:
   A. Create a new user in the Manage Users and Groups portlet to replace the current WebSphere Application Server administrative user.
   B. Create a new user in the Manage Users and Groups portlet to replace the current WebSphere Portal administrative user.
   C. Create a new group in the Manage Users and Groups portlet to replace the current group.
   D. Run the `./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task, from the `wp_profile_root/ConfigEngine` directory, to replace the old WebSphere Application Server administrative user ID and group ID with the new user and group. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.
   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.
   E. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.
   F. Run the `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task to replace the old WebSphere Portal administrative user ID and group ID with the new user and group. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.
   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.
   G. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

11. Optional: Perform the following steps to set the realm you created as the default realm. **Remember:** Only users defined in base entries that exist in the default realm are able to log into WebSphere Portal. If you find that a user cannot log in to WebSphere Portal, check to see if the base entry that contains the user exists in the default realm. You can run the `wp-query-realm-baseentry` task to see what base entries are part of the default realm. If the default realm is missing
the base entry, run the \texttt{wp-add-realm-baseentry} task to add the base entry to the default realm.

A. Use a text editor to open the \texttt{wkplc.properties} file, located in the \texttt{wp_profile_root}/ConfigEngine/properties directory.

B. For \texttt{defaultRealmName}, type the \texttt{realmName} property value you want to use as the default realm.

C. Save your changes to the \texttt{wkplc.properties} file.

D. Run the \texttt{./ConfigEngine.sh wp-default-realm -DWasPassword=password} task, from the \texttt{wp_profile_root}/ConfigEngine directory, to set this realm as the default realm.

E. Stop and restart all necessary servers to propagate your changes.

12. Optional: Perform the following steps to query a realm for a list of its base entries:

A. Use a text editor to open the \texttt{wkplc.properties} file, located in the \texttt{wp_profile_root}/ConfigEngine/properties directory.

B. For \texttt{realmName}, type the name of the realm you want to query.

C. Save your changes to the \texttt{wkplc.properties} file.

D. Run the \texttt{./ConfigEngine.sh wp-query-realm-baseentry -DWasPassword=password} task, from the \texttt{wp_profile_root}/ConfigEngine directory, to list the base entries for a specific realm.

13. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

A. Use a text editor to open the \texttt{wkplc.properties} file, located in the \texttt{wp_profile_root}/ConfigEngine/properties directory.

B. Enter a value for \texttt{realmName} or leave blank to update the default realm.

C. Save your changes to the \texttt{wkplc.properties} file.

D. Run the \texttt{./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password} task, located in the \texttt{wp_profile_root}/ConfigEngine directory, to enable the distinguished name login. \textbf{Note:} After running this task to enable the full distinguished name login, you can run the \texttt{./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password} task to disable the feature.

E. Stop and restart all necessary servers to propagate your changes.

If you created your clustered environment then performed the steps in this task, you must now run the \texttt{enable-jcr-security} task on the secondary node. See \textit{Enabling LDAP security after cluster creation} for instructions.

\textbf{Parent topic:} Configuring the default federated repository on AIX in a clustered environment

\textbf{Related tasks}

Starting and stopping servers, deployment managers, and node agents

Enabling LDAP security after cluster creation

\begin{center}
\begin{tabular}{|c|c|}
\hline
1253
\hline
\end{tabular}
\end{center}
Adapting the attribute configuration

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

After installation, IBM WebSphere Portal has a predefined set of attributes for users and groups. Your LDAP server may have a different set of predefined user and group attributes. To ensure proper communication between WebSphere Portal and your LDAP server, you can configure additional attributes and flag existing attributes as required or unsupported on a per repository basis or for all configure repositories.

LDAP servers can only handle attributes that are explicitly defined in their schema. The LDAP server’s schema is different from the WebSphere Portal schema but the two schemas should match for proper communication between WebSphere Portal and the LDAP server. The task to add the LDAP user registry does some basic attribute configurations depending on the type of LDAP server that you choose. You may, however, still need to adapt the WebSphere Portal configuration to match the LDAP schema; for example, if an attribute is defined in WebSphere Portal but not in the LDAP server, you will need to perform one of the following tasks to resolve this mismatch:

- Flag the attribute as unsupported for the LDAP server
- Introduce an attribute mapping that maps the WebSphere Portal attribute to an attribute defined in the LDAP schema

Perform the following tasks to adapt the attribute configuration to match the configured LDAP server(s) and your business needs:

1. **Querying the defined attributes on AIX**
   After installing IBM WebSphere Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

2. **Adding attributes on AIX in a clustered environment**
   The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM WebSphere Portal and your user registry.

3. **Mapping attributes on AIX in a clustered environment**
   After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

4. **Removing attributes**
   Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

**Parent topic:** Choosing your user registry model on AIX in a clustered environment

**Related tasks**
- Adding an LDAP user registry on AIX
- Adding an LDAP user registry over SSL on AIX
- Configuring a stand-alone LDAP user registry on AIX
Configuring a stand-alone LDAP user registry over SSL on AIX
Querying the defined attributes on AIX

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

**Warning:** If you are using a database user registry or a property extension database, copy the database drivers to the `AppServer_root/lib` directory before executing this script.

Run the `./ConfigEngine.sh wp-query-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, any time during the configuration process or at runtime to query an overview of the currently defined attributes. This task creates the `availableAttributes.html` report, located in the `wp_profile_root/ConfigEngine/log` directory. The report contains one table that lists the available attributes for Users (**PersonAccount**) and one table that lists the available attributes for Groups. For each configured repository there is a column that indicates if the attribute is flagged as unsupported or if the attribute is mapped to a different LDAP attribute. **Note:** This task does not validate the existence of attributes in the LDAP schema.

**Parent topic:** Adapting the attribute configuration

**Next topic:** Adding attributes on AIX in a clustered environment
Adding attributes on AIX in a clustered environment

The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM® WebSphere® Portal and your user registry.

Perform the following steps to add new attributes to your user registry:

1. Install the required Enterprise Archive (.ear) file on WebSphere Application Server. Select the appropriate task to install the .ear file: Table 1. Steps for installing the .ear file by environment

<table>
<thead>
<tr>
<th>Environment</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone environment</td>
<td>Perform the following steps to install the .ear file in a stand-alone environment: Open a command prompt. Navigate to the <code>wp_profile_root/ConfigEngine</code> directory. Run the <code>./ConfigEngine.sh wp-la-install-ear -DWasPassword=password</code> task.</td>
</tr>
<tr>
<td>Cluttered environment</td>
<td>Perform the following steps to install the .ear file in a clustered environment: Open a command prompt. Navigate to the <code>wp_profile_root/ConfigEngine</code> directory on the primary node. Run the <code>./ConfigEngine.sh wp-la-install-ear -DWasPassword=dmgr_password -DServerName=dmgr -DNodeName=node_name</code> task. Where <code>node_name</code> is the name of the node where the deployment manager resides; you can find the <code>node_name</code> value in the WebSphere Application Server Administrative Console under <code>System administrator &gt; Deployment Manager &gt; Runtime tab &gt; General Properties &gt; Node Name</code>.</td>
</tr>
</tbody>
</table>

2. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

3. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

4. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Property Extension Properties heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - `la.providerURL`
   - `la.propertyName`
   - `la.entityTypes`
   - `la.dataType`
   - `la.multiValued`

5. Save your changes to the `wkplc.properties` file.

6. Run the `./ConfigEngine.sh wp-add-property -DWasPassword=password` task to add the attribute to the user registry.

   Note: This task performs an EJB call to WebSphere Application Server, which must authenticate against WebSphere.
Application Server. Depending on the configuration in the `sas.client.props` file, you may receive a popup window or a command line prompt asking for user identity and password. Enter the WebSphere Application Server user ID and password.

**Remember:** If you have multiple properties to add, repeat all steps, except for the `wp-la-install-ear` task, until all new attributes are added.

7. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents.*

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Adapting the attribute configuration

**Previous topic:** Querying the defined attributes on AIX

**Next topic:** Mapping attributes on AIX in a clustered environment

**Related tasks**
- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation
Mapping attributes on AIX in a clustered environment

After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

Perform the following steps to map attributes between WebSphere Portal and your LDAP server; if you have multiple LDAP servers, you will need to perform these steps for each LDAP server:

1. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

2. Enter a value for one of the following sets of parameters in the `wkplc.properties` file to identify your LDAP server:
   - **Note:** Make sure you use the same values you used to configure your LDAP server.
   
   Table 1. Identifying your LDAP server in the `wkplc.properties` file.

3. Run one of the following tasks to check that all defined attributes are available in the configured LDAP user registry:
   - **/ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password task, from the `wp_profile_root/ConfigEngine` directory.
   - **/ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password task, from the `wp_profile_root/ConfigEngine` directory.

4. Open the `ConfigTrace.log` file, located in the `wp_profile_root/ConfigEngine/log` directory, to review the following output for the **PersonAccount** and **Group** entity type:
- The following attributes are defined in WebSphere Portal but not in the LDAP server
  - This list contains all attributes that are defined in WebSphere Portal but not available in the LDAP. Flag attributes that you do not plan to use in WebSphere Portal as unsupported. Map the attributes that you plan to use to the attributes that exist in the LDAP; you must also map the uid, cn, firstName, sn, preferredLanguage, and ibm-primaryEmail attributes if they are contained in the list.

- The following attributes are flagged as required in the LDAP server but not in WebSphere Portal
  - This list contains all attributes that are defined as "MUST" in the LDAP server but not as required in WebSphere Portal. You should flag these attributes as required within WebSphere Portal; see the step below about flagging an attribute as either unsupported or required.

- The following attributes have a different type in WebSphere Portal and in the LDAP server
  - This list contains all attributes that WebSphere Portal might ignore because the data type within WebSphere Portal and within the LDAP server do not match.

5. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

6. Enter a value for one of the following sets of parameters in the wkplc.properties file to correct any issues found in the config trace file: Table 3. Parameters to define in the wkplc.properties file to correct any issues found in the config trace file.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>The following parameters are found under the LDAP attribute configuration heading: <strong>Note:</strong> See the properties file for specific information about the required parameters and for advanced parameters. standalone.idstandalone.ldap.attributes.nonSupportedstandalone.ldap.attributes.nonSupported.deletestandalone.ldap.attributes.mapping.ldapNamestandalone.ldap.attributes.mapping.portalNamestandalone.ldap.attributes.mapping.entityTypes For example, the following values will flag certificate and members as unsupported attributes and will map ibm-primaryEmail to mail and ibm-jobTitle to title for both the PersonAccount and Group entityTypes: standalone.ldap.attributes.nonSupported=certificate, members standalone.ldap.attributes.nonSupported.delete standalone.ldap.attributes.mapping.ldapNamemail, title standalone.ldap.attributes.mapping.portalName=ibm-primaryEmail, ibm-jobTitle standalone.ldap.attributes.mapping.entityTypes=PersonAccount, Group</td>
</tr>
</tbody>
</table>
7. Save your changes to the `wkplc.properties` file.
8. Run one of the following tasks to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between WebSphere Portal and the LDAP user registry:

   Table 4. Task to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between Portal and the LDAP user registry.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td><code>./ConfigEngine.sh wp-update-standalone-ldap-attribute-config -DWasPassword=password</code> task, from the <code>wp_profile_root/ConfigEngine</code> directory</td>
</tr>
<tr>
<td>Federated</td>
<td><code>./ConfigEngine.sh wp-update-federated-ldap-attribute-config -DWasPassword=password</code> task, from the <code>wp_profile_root/ConfigEngine</code> directory</td>
</tr>
</tbody>
</table>

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: `Starting and stopping servers, deployment managers, and node agents`.

10. Optional: Perform the following steps to flag an attribute as either unsupported or required for the entire WebSphere Portal environment instead of just for the specified LDAP:

   A. Enter a value for the following required parameters in the `wkplc.properties` file: 
      
      ```
      user.attributes.required
      user.attributes.nonsupported
      ```
      
      ```Note:` See the properties file for specific information about the required parameters and for advanced parameters.

   B. Save your changes to the `wkplc.properties` file.

   C. Run the `./ConfigEngine.sh wp-update-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory.

   D. Stop and restart all necessary servers to propagate your changes.

   If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See `Enabling LDAP security after cluster creation` for instructions.
Parent topic: Adapting the attribute configuration
Previous topic: Adding attributes on AIX in a clustered environment
Next topic: Removing attributes

Related tasks
Starting and stopping servers, deployment managers, and node agents
Enabling LDAP security after cluster creation
Removing attributes

Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

Perform the following steps to remove an attribute from your database:

**Important:** Do not remove attributes that have already been populated with user values because this can cause database inconsistencies.

**Cluster note:** In a clustered environment, perform the following steps on the deployment manager and then resynch the nodes.

1. Perform the following steps to remove an attribute stored in a property extension database; this is an attribute that was added using the `wp-add-la-property` task.

   A. Open the tool you use to edit your database.
   B. Verify that your attribute name is available in the LAPROP table.
   C. Delete the required attributes from the LAPROP table.
   D. Open the `wimxmlextension.xml` file, located in the `wp_profile_root/config/cells/cellname/wim/model` directory.
   E. Locate and delete the `propertySchema` definition for the attributes that you deleted from the LAPROP table; for example:

   ```xml
   <wim:propertySchema
       nsURI="http://www.ibm.com/websphere/wim"
       dataType="String"
       multiValued="true"
       propertyName="attribute_name">
     <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
   </wim:propertySchema>
   ```
   F. Save your changes to the `wimxmlextension.xml` file.
   G. Open the `wimconfig.xml` file, located in the `wp_profile_root/config/cells/cellname/wim/config` directory.
   H. Locate and delete the `propertiesNotSupported` definitions for the attributes that you deleted from the LAPROP table; for example:

   ```xml
   <config:propertiesNotSupported name="attribute_name"/>
   ```
   I. Save your changes to the `wimconfig.xml` file.
   J. Stop and restart the server1 and WebSphere_Portal servers from the `wp_profile_root/bin` directory.

2. Perform the following steps to remove an attribute that is not stored in a property extension database:

   A. Open the `wimxmlextension.xml` file.
   B. Locate and delete the `propertySchema` definition for the attributes you previously added; for example:

   ```xml
   <wim:propertySchema
       nsURI="http://www.ibm.com/websphere/wim"
       dataType="String"
       multiValued="true"
       propertyName="attribute_name">
     <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
   </wim:propertySchema>
   ```
   C. Save your changes to the `wimxmlextension.xml` file.
   D. Open the `wimconfig.xml` file.
   E. Locate and delete the stanza that corresponds to the custom attribute you deleted from the `wimextension.xml` file; for example:

   ```xml
   <config:attributes
       name="attribute_name"
       propertyName="property_name"/>
   ```
   F. Save your changes to the `wimconfig.xml` file.
G. Stop and restart the server1 and WebSphere_Portal servers.

**Parent topic:** Adapting the attribute configuration

**Previous topic:** Mapping attributes on AIX in a clustered environment
Preparing additional nodes on AIX

After installing and configuring your primary node, you can create your secondary nodes. You must install IBM® WebSphere® Portal on each node and then configure the node to access the database and user registry before adding it to the cluster.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on AIX
  - Preparing your AIX operating system
  - Preparing the primary node on AIX
  - Choosing the type of cluster to create on AIX
  - Installing a secondary node after migrating from previous version

Perform the following tasks to prepare your secondary node:

1. **Installing WebSphere Portal on AIX on the additional nodes**
   Install IBM WebSphere Portal on your secondary nodes to create a highly available and scalable environment.

2. **Choosing the type of additional node to create on AIX**
   If you created a static cluster using the IBM WebSphere Application Server Network Deployment, add additional static nodes to the cluster. If you created a dynamic cluster using IBM WebSphere Virtual Enterprise, add additional dynamic nodes to the cluster.

3. **Choosing the type of vertical cluster to create on AIX**
   If you are using the IBM WebSphere Application Server Network Deployment, you can add vertical cluster members to your static cluster. If you are using IBM WebSphere Virtual Enterprise, you can add vertical cluster members to your dynamic cluster.

**Parent topic:** Setting up a cluster on AIX

**Previous topic:** Configuring WebSphere Portal to use a user registry on AIX in a clustered environment

**Next topic:** Tune your servers
Installing WebSphere Portal on AIX on the additional nodes

Install IBM® WebSphere® Portal on your secondary nodes to create a highly available and scalable environment.

Review the Supported hardware and software requirements for your operating system and ensure that all hardware and software matches the minimum product level before installing WebSphere Portal. Review Installation methods, options, and sources before installing to determine your installation source location and installation method. **Note:** If you plan to use IBM WebSphere Application Server Version 7.0, you must purchase and install it before installing WebSphere Portal on the existing WebSphere Application Server Version 7.0 instance.

Perform the following steps to install WebSphere Portal:

1. **Type ping yourserver.yourcompany.com on a command line to verify that your fully qualified host name is properly configured.**

2. **If you are installing with an existing WebSphere Application Server instance, ensure it is installed at the supported level and has the following features installed:**
   - **Restriction:** Installing WebSphere Portal Version 6.1.0.1 or higher on an existing WebSphere Application Server Version 7.0 instance where WebSphere Portal Version 7.0 was installed is unsupported.
   - **Important:** Besides ensuring that the existing WebSphere Application Server instance is at the supported level, you will also need to ensure that Apache Derby is installed at the supported level before installing WebSphere Portal. See WebSphere Portal detailed system requirements for supported levels.

   - Application Server
   - Administration
     - Scripted Administration
     - Administrative Console
   - Ant and Deployment Tools
     - Deploy Tool
     - Ant Utilities

3. **Optional:** Perform the following steps if you want to install WebSphere Portal using a non-root user:
   - **Restriction:** Although it is possible to install WebSphere Application Server as a non-root user, there are limitations to this option that can affect WebSphere Portal functionality, which is why you should install WebSphere Application Server as a root user.
   - **A.** Install the current supported version of WebSphere Application Server as a root user.
   - **B.** Open a command prompt and perform the following steps to create a non-root user and to change ownership:

   1. Use the appropriate system commands to create a new group, to create a new user, and to add the new user to the new group.
   2. Run the following tasks to change the rights of the non-root user:

   ```
   chmod -R g+rwx /opt/IBM
   chgrp -R group_name /opt/IBM
   chmod -R g+wr /tmp
   chgrp -R group_name /tmp
   chmod -R g+wr /var/tmp
   ```
C. Launch the installation program per the next step. During the installation, select the existing WebSphere Application Server using the non-root user and set the installation location to a unique location under the path where non-root permissions were granted.

4. Choose one of the following installation commands based on the installation method you want to use to install WebSphere Portal; see Installation Methods for more information:

**Advance installation parameters:** To customize your installation, you can add various parameters to your installation commands; for example, you can specify your port information. See Advanced installation parameters for information.

**Restriction:** When defining your installation path, the **ONLY** supported characters are ASCII letters [A-Z and a-z], numbers [0-9], slashes [/ or \, depending on your operating system], and the dash [-].

### Option Description

<table>
<thead>
<tr>
<th>Graphical user interface</th>
<th>./install.sh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Console mode</td>
<td>./install.sh -console from the PortalExpress subdirectory</td>
</tr>
<tr>
<td>Silent Install</td>
<td>./install.sh -options &quot;path_to_file/response_filename&quot;, where path_to_file is the full path to the response file and response_filename is the name of the file. A sample install response file (installresponse.txt) and a sample uninstall response file (uninstallresponse.txt) are located in the setup CD root directory. <strong>Important:</strong> Do not place the response file in a path that contains a space and do not put a space in the file name.</td>
</tr>
</tbody>
</table>

5. Verify your installation was successful; access WebSphere Portal using the http://yourserver:yourport/wps/portal format. Run the ../ConfigEngine.sh list-server-ports -DWasPassword=password task from the wp_profile_root /ConfigEngine directory to generate the wp_profile_root/ConfigEngine/log/wp_PortMatrix.txt file that lists WebSphere Portal ports for your installation.

6. Optional: Perform the following steps if you need to change the ports for WebSphere Application Server or WebSphere Portal:

**Note:** The starting port parameter is required for a successful completion of the modify-ports-by-startport task. Once you specify a start port, this port becomes the base for assigning port values. The code increments this value as each port is assigned, which means that the WebSphere Portal ports will be assigned incrementally starting with the port defined with the modify-ports-by-startport task.

A. Stop the server1 and WebSphere_Portal servers.

B. Run one of the following commands for each server you need to change:

<table>
<thead>
<tr>
<th>Method</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting port number</td>
<td>.ConfigEngine.sh modify-ports-by-startport -DWasPassword=password -DModifyPortsServer=servername -DStartPort=starting port number</td>
</tr>
</tbody>
</table>
C. Restart the server1 and WebSphere_Portal servers.

7. Perform the following steps if you migrated from a previous version of WebSphere Portal:
   A. Log on to the WebSphere Application Server Administrative Console.
   C. Click WP Identification.
   D. Click Custom properties.
   E. If applicable, note the location of the LpidToGupidMapping.properties file.
   F. If the LpidToGupidMapping.properties file exists on the primary node, copy the file to the same location on the secondary node.

**Port fileNote:** Sample port files are available on the Setup disc.

```bash
./ConfigEngine.sh modify-ports-by-portsfile -DWasPassword=password -DModifyPortsServer=servername -DPortsFile=full path to ports file
```

The following is an example of the information within a port file although the port values will be different based on your environment:

```
BOOTSTRAP_ADDRESS=10031
SOAP_CONNECTOR_ADDRESS=10033
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=10032
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=10025
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=10036
WC_adminhost=10027
WC_defaulthost=33344
DCS_UNICAST_ADDRESS=10029
WCadminhost_secure=10039
WCdefaulthost_secure=10035
SIB_ENDPOINT_ADDRESS=10026
SIB_ENDPOINT_SECURE_ADDRESS=10037
SIB_MQ_ENDPOINT_ADDRESS=10030
SIB_MQ_ENDPOINT_SECURE_ADDRESS=10028
ORB_LISTENER_ADDRESS=10034
```

**Parent topic:** Preparing additional nodes on AIX

**Next topic:** Choosing the type of additional node to create on AIX

**Related concepts**

- Installation methods
- IBM Support Assistant Lite for WebSphere Portal

**Related reference**

- Advanced installation parameters
Choosing the type of additional node to create on AIX

If you created a static cluster using the IBM® WebSphere® Application Server Network Deployment, add additional static nodes to the cluster. If you created a dynamic cluster using IBM WebSphere Virtual Enterprise, add additional dynamic nodes to the cluster.

Choose one of the following options to add additional nodes to your cluster:

- **Adding additional nodes to the static cluster on AIX**
  After installing IBM WebSphere Portal on your secondary node, you must add the node to the cluster to create a highly available environment.

- **Adding an additional node to an existing dynamic cluster**
  After creating your dynamic cluster, you can add additional nodes to expand the capacity of the dynamic cluster. Install and configure the additional node, then add it to the existing dynamic cluster node group, and then add it to the existing IBM WebSphere Virtual Enterprise dynamic cluster.

Parent topic: Preparing additional nodes on AIX

Previous topic: Installing WebSphere Portal on AIX on the additional nodes

Next topic: Choosing the type of vertical cluster to create on AIX
Adding additional nodes to the static cluster on AIX

After installing IBM® WebSphere® Portal on your secondary node, you must add the node to the cluster to create a highly available environment.

Perform the following steps to add the secondary node to the cluster:

1. Perform the following steps on the secondary node to access your database server: **Note**: All properties files are located in the `{wp_profile_root}/ConfigEngine/properties` directory.
   
   A. For JDBC Type 2 connections only, ensure that the database client software is installed on the secondary node using the same settings as the primary node and that you can connect to the remote database. It is not necessary to create databases or users when configuring secondary nodes. **DB2 users**: When using DB2, you must catalog the TCP/IP node and all databases used on each secondary node. For more information refer to Preparing DB2.
   
   B. For JDBC Type 4 connections only, copy the JDBC Type 4 jar files to the secondary node from the DB2 server.
   
   C. Set all database properties in the `{wkplc_dbtype.properties}` and `{wkplc_comp.properties}` files with the appropriate values from the primary node.

2. Run the following tasks to validate your database settings:
   
   A. `./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password`
   
   B. `./ConfigEngine.sh validate-database-connection -DWasPassword=password -DTransferDomainList=release,customization,community,jcr,feedback,likeminds`

3. Stop the server1 and WebSphere_Portal servers on the secondary node and ensure that the following parameters are set correctly in the `{wkplc.properties}` file, located in the `{wp_profile_root}/ConfigEngine\properties` directory of the secondary node: **Note**: Although you can add these parameters directly to any task that you run while creating your cluster, you may want to add them to the properties file while creating your cluster and then remove them when you are finished to keep your environment secure.
   
   A. Verify that `WasUserid` is set to your deployment manager administrator ID.
   
   B. Verify that `WasPassword` is set to your deployment manager administrator password.
   
   C. Verify that `PortalAdminId` is set to your WebSphere Portal administrator ID.
   
   D. Verify that `PortalAdminPwd` is set to your WebSphere Portal password.
   
   E. Verify that `WasRemoteHostName` is set to the fully qualified host name of the deployment manager.
   
   F. Verify that `WasSoapPort` is set to the SOAP port that the deployment manager is using; the default value is 8879.
   
   G. Verify that `PrimaryNode` is set to `false`.
   
   H. Verify that `ClusterName` is set to the primary node’s `ClusterName`.

4. Run the `./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=dmgr_password` task, from the `{wp_profile_root}/ConfigEngine` directory of the secondary node.

5. Optional: If you configured a database user registry or a property extension (lookaside) database, run the following task to set up access to the database drivers:
   
   **Note**: You will also need to run this task if you migrated the primary node from a release prior to Version 6.1.0, even if you did not configure a database user registry or a property extension database. (`la.DbType` is not set for a property extension database in the `{wkplc.properties}` file).
   
   A. Set the property value for `federated.db.DbType` if using a database user registry or set the property value for `la.DbType` if using a property extension database in the `{wkplc.properties}` file.
   
   **Note**: If you migrated the primary node from a version prior to Version 6.1.0 and you are not using a database user registry or a property extension database, set the property value for `federated.db.DbType` to the same value that is...
in the wkplc.properties file on the primary node.

B. Run the .\ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=dmgr_password - DDdDomain=ls|federated.db -DVmmNodeName=node_name -Ddb_type.NodeDbLibrary=local full path of the database jars task from the wp_profile_root/ConfigEngine directory to create the variable used to access the VMM database jars.

**Note:** VmmNodeName is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The db_type in db_type.NodeDbLibrary should be set to the type of database you are using, for example db2.

6. Run the .\ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=dmgr_password task. **Note:** If you want to customize the WebSphere Portal server name (WebSphere_Portal) for your additional node, set the ServerName parameter in the wkplc.properties file after running the cluster-node-config-post-federation task.

7. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell's user registry. Run the .\ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password - DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid task, from the wp_profile_root/ConfigEngine directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

**Attention:** The password value for -DWasPassword is the Deployment Manager administrative password.

**Important:** You must provide the full distinguished name (DN) for the newAdminId and newAdminGroupId parameters.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the -Dskip.ldap.validation=true parameter to the task to skip the validation.

**Fast path:** If the value for newAdminGroupId contains a space; for example Software Group, open the wkplc.properties file and add the values for newAdminId, newAdminPw, and newAdminGroupId. Save your changes and run the ConfigEngine.bat wp-change-portal-admin-user -DWasPassword=dmgr_password task.

**Important:** If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the wp-change-portal-admin-user task until after the cluster-node-config-cluster-setup (static cluster) or cluster-node-config-dynamic-cluster-setup (dynamic cluster) task completes. After running the wp-change-portal-admin-user task, start or restart the WebSphere_Portal server to use the updated administrator user ID.

**Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the wp-change-portal-admin-user task. -DnewAdminPw is an optional parameter to update the Administrative password in the wkplc.properties file if required.

8. Optional: Open the wkplc.properties file and change the ServerName parameter from the default WebSphere_Portal_node_value to a value that meets your business needs. Do not change the parameter to WebSphere_Portal as this is the primary node value.

9. Run the .\ConfigEngine.sh cluster-node-config-cluster-setup -DWasPassword=dmgr_password task to add the node to your cluster.

10. Wait at least 30 minutes before stopping and restarting the deployment manager because the node agent runs a background thread to expand all EAR files into their target directories. **Note:** Attempting to restart the deployment manager while the expansion is in process can result in an unusable application. Although the time to complete an EAR expansion is environment-dependent, a 30 minute wait is sufficient for a typical environment.

11. Perform the following steps to access the Web Content Management content through an external Web server:

A. Log on to the deployment manager administrative console.

B. Select Environment > WebSphere Variables.

C. From the Scope drop-down menu, select the Node-node_name, Server-servername option to narrow the scope of the listed variables, where Node-node_name is the node that contains the application server.
D. Update the **WCM_HOST** variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.

E. Update the **WCM_PORT** variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.

F. Perform the following steps to synchronize the node with the deployment manager:
   1. Select **System Administration > Nodes**.
   2. Select the node that you want to synchronize from the list.
   3. Click **Full Resynchronize**.

G. Log off of the deployment manager administrative console.

12. If you entered passwords in any of the properties files while creating your cluster, you should remove them for security purposes. See "Deleting passwords from properties files" under Related tasks for information.

**Parent topic:** Choosing the type of additional node to create on AIX

**Related tasks**

Deleting passwords from properties files
Adding an additional node to an existing dynamic cluster

After creating your dynamic cluster, you can add additional nodes to expand the capacity of the dynamic cluster. Install and configure the additional node, then add it to the existing dynamic cluster node group, and then add it to the existing IBM® WebSphere® Virtual Enterprise dynamic cluster.

On all WebSphere Portal nodes, install WebSphere Virtual Enterprise and augment the wp_profile profile. See Installing and configuring the product for information about installing WebSphere Virtual Enterprise. See Augmenting profiles for information about augmenting the Deployment Manager and wp_profile profiles.

**Prerequisites**

- PK97393 for all WebSphere Application Server Versions prior to 6.1.0.5
- Installing and configuring the product:
  - Augmenting profiles
  - PK97393

Perform the following steps to add an additional node to an existing WebSphere Virtual Enterprise dynamic cluster:

1. Perform the following steps on the secondary node to access your database server. **Note**: All properties files are located in the `wp_profile_root/ConfigEngine/properties` directory.
   A. For JDBC Type 2 connections only, ensure that the database client software is installed on the secondary node using the same settings as the primary node and that you can connect to the remote database. It is not necessary to create databases or users when configuring secondary nodes. **DB2 users**: When using DB2, you must catalog the TCP/IP node and all databases used on each secondary node. For more information refer to Preparing DB2.
   B. For JDBC Type 4 connections only, copy the JDBC Type 4 jar files to the secondary node from the DB2 server.
   C. Set all database properties in the `wkplc_dbtype.properties` and `wkplc_comp.properties` files with the appropriate values from the primary node.

2. Run the following tasks to validate your database settings:
   A. `./ConfigEngine.sh validate-database-driver`  
      
      DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   
   B. `./ConfigEngine.sh validate-database-connection` -DWasPassword=password  
      
      DTransferDomainList=release,customization,community,jcr,feedback,likeminds

3. Stop the server1 and WebSphere_Portal servers on the secondary node and ensure that the following parameters are set correctly in the `wkplc.properties` file, located in the `wp_profile_root\ConfigEngine\properties` directory of the secondary node. **Note**: Although you can add these parameters directly to any task that you run while creating your cluster, you may want to add them to the properties file while creating your cluster and then remove them when you are finished to keep your environment secure.
   A. Verify that `WasUserid` is set to your deployment manager administrator ID.
   B. Verify that `WasPassword` is set to your deployment manager administrator password.
   C. Verify that `PortalAdminId` is set to your WebSphere Portal administrator ID.
   D. Verify that `PortalAdminPwd` is set to your WebSphere Portal password.
   E. Verify that `WasRemoteHostName` is set to the fully qualified host name of the deployment manager.
   F. Verify that `WasSoapPort` is set to the SOAP port that the deployment manager is using; the default value is `8879`.
   G. Verify that `PrimaryNode` is set to `false`.  

1273
H. Verify that **ClusterName** is set to the primary node’s **ClusterName**.

4. Run the `./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=dmgr_password` task, from the `wp_profile_root/ConfigEngine` directory of the secondary node.

5. Optional: If you configured a database user registry or a property extension (lookaside) database, run the following task to set up access to the database drivers:

   **Note:** You will also need to run this task if you migrated the primary node from a release prior to Version 6.1.0, even if you did not configure a database user registry or a property extension (lookaside) database.

   A. Set the property value for **federated.db.DbType** if using a database user registry or set the property value for **la.DbType** if using a property extension database in the `wkplc.properties` file.

      **Note:** If you migrated the primary node from a version prior to Version 6.1.0 and you are not using a database user registry or a property extension database, set the property value for **federated.db.DbType** to the same value that is in the `wkplc.properties` file on the primary node.

   B. Run the `./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=dmgr_password -DDbDomain=la|federated.db -DVmmNodeName=node_name -Db_type.NodeDbLibrary=local full path of the database jars` task from the `wp_profile_root/ConfigEngine` directory to create the variable used to access the VMM database jars.

      **Note:** **VmmNodeName** is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The **db_type** in **Db_type.NodeDbLibrary** should be set to the type of database you are using, for example db2.

6. Run the `./ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=dmgr_password` task. **Note:** If you want to customize the WebSphere Portal server name (WebSphere_Portal) for your additional node, set the **ServerName** parameter in the `wkplc.properties` file after running the `cluster-node-config-post-federation` task.

7. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell’s user registry. Run the `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task, from the `wp_profile_root/ConfigEngine` directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

   **Attention:** The password value for **-DWasPassword** is the Deployment Manager administrative password. **Important:** You must provide the full distinguished name (DN) for the **newAdminId** and **newAdminGroupId** parameters.

   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the **-Dskip.ldap.validation=true** parameter to the task to skip the validation.

   **Fast path:** If the value for **newAdminGroupId** contains a space; for example Software Group, open the `wkplc.properties` file and add the values for **newAdminId**, **newAdminPw**, and **newAdminGroupId**. Save your changes and run the `ConfigEngine.bat wp-change-portal-admin-user -DWasPassword=dmgr_password` task.

   **Important:** If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user` task until after the `cluster-node-config-cluster-setup (static cluster) or cluster-node-config-dynamic-cluster-setup (dynamic cluster)` task completes. After running the `wp-change-portal-admin-user` task, start or restart the WebSphere_Portal server to use the updated administrator user ID.

   **Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the `wp-change-portal-admin-user` task. **-DnewAdminPw** is an optional parameter to update the Administrative password in the `wkplc.properties` file if required.

8. Log on to the deployment manager administrative console.

9. Perform the following steps to add members to the node group:

   A. Click **System administration > Node groups**.
B. Click on the name of the node group that you want to add members to.
C. Click **Node group members** under Additional Properties.
D. Click **Add**.
E. Select the additional node you want to add into the node group and then click **Add**.
F. Click the **Save** link to save your changes to the master configuration.

10. Wait at least 30 minutes before stopping and restarting the deployment manager because the node agent runs a background thread to expand all EAR files into their target directories. **Note:** Attempting to restart the deployment manager while the expansion is in process can result in an unusable application. Although the time to complete an EAR expansion is environment-dependent, a 30 minute wait is sufficient for a typical environment.

11. Define or verify the following parameters in the **wkplc.properties** file, located in the **wp_profile_root**/ConfigEngine/properties directory:
   
   A. Set **ClusterName** to the name of the existing dynamic cluster.
   B. Verify that **CellName** is set to the deployment manager cell.
   C. Verify that **NodeName** is set to the local WebSphere Portal node.
   D. Set **ServerName** to the server that will be used for the dynamic cluster member on this node. **Note:** Log on to the deployment manager administrative console and click **Dynamic Clusters > PortalCluster > Dynamic cluster members** to find the name of the server used for the dynamic cluster.
   E. Verify that **PrimaryNode** is set to **false** because this is an additional node.

12. Run the **./ConfigEngine.sh cluster-node-config-dynamic-cluster-setup -DWasPassword=dmgr_password** task to add the new member to the existing dynamic cluster. **Note:** This task may display a warning message since the cluster already exists but it will proceed to create the new dynamic cluster; therefore, the warning can be ignored.

13. Perform the following steps to start the cluster member:
   
   A. Log on to the Deployment Manager administrative console.
   B. Navigate to **Servers > Dynamic clusters > cluster name > Dynamic cluster members**.
   C. Select the cluster member and click **Start**.
   D. Log off of the deployment manager administrative console.

14. Perform the following steps to access the Web Content Management content through an external Web server:
   
   A. Log on to the deployment manager administrative console.
   B. Select **Environment > WebSphere Variables**.
   C. From the **Scope** drop-down menu, select the **Node=nodename, Server=servername** option to narrow the scope of the listed variables, where **Node=nodename** is the node that contains the application server.
   D. Update the **WCM_HOST** variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.
   E. Update the **WCM_PORT** variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.
   F. Perform the following steps to synchronize the node with the deployment manager:
      1. Select **System Administration > Nodes**.
      2. Select the node that you want to synchronize from the list.
      3. Click **Full Resynchronize**.
   G. Log off of the deployment manager administrative console.

15. Run the following tasks to propagate your changes. **Note:** **WebSphere_Portal_nodename** is the name of the node’s WebSphere Portal server; but if you customized the server name, the default name changes. If you are uncertain of the server name, run the **serverstatus -all** task to get a list of the server names and their status.

   A. **./stopManager.sh -username admin_userid -password admin_password**, from the **dmgr_profile_root\bin** directory
   B. **./stopNode.sh -username admin_userid -password admin_password**, from the **wp_profile_root\bin** directory
   C. **./stopServer.sh WebSphere_Portal_nodename -username admin_userid -password admin_password**
D. 
E. 
16. If you entered passwords in any of the properties files while creating your cluster, you should remove them for security purposes. See "Deleting passwords from properties files" under Related tasks for information.

Parent topic: Choosing the type of additional node to create on AIX

Related tasks
Deleting passwords from properties files
Choosing the type of vertical cluster to create on AIX

If you are using the IBM® WebSphere® Application Server Network Deployment, you can add vertical cluster members to your static cluster. If you are using IBM WebSphere Virtual Enterprise, you can add vertical cluster members to your dynamic cluster.

Choose one of the following options to add vertical cluster members to your cluster.

- **Adding vertical cluster members to a static cluster on AIX**
  You can add vertical cluster members to share the workload demands of your cluster across multiple members running on the same physical machine.

- **Adding vertical cluster members to a dynamic cluster on AIX**
  You can add vertical cluster members to share the workload demands of your cluster across multiple members running on the same physical machine.

**Parent topic:** Preparing additional nodes on AIX

**Previous topic:** Choosing the type of additional node to create on AIX
Adding vertical cluster members to a static cluster on AIX

You can add vertical cluster members to share the workload demands of your cluster across multiple members running on the same physical machine.

Perform the following steps to add a vertical cluster member to your clustered environment:

1. Log into the deployment manager administrative console.

2. Depending on your version of WebSphere Application Server, make the appropriate selection from the navigation:
   - For WebSphere Application Server Version 6.1: Click Servers > Clusters in the console navigation tree, select the cluster name, and then click Cluster members from the list of additional properties.
   - For WebSphere Application Server Version 7.0: Click Servers > Clusters > WebSphere application server clusters > cluster_name > Cluster members.

3. Click New to create the cluster member.
   A. Define the name of cluster member. **Note:** Do not use spaces in the cluster member name.
   B. Select an existing node where IBM® WebSphere® Portal is installed.
   C. Check the box Generate Unique HTTP Ports.
   D. Click Add Member and then click Next to view the summary.
      **Important:** You must update the virtual host entries for the new port created when adding a cluster member. You can do this by updating the default_host virtual host in the administrative console and adding a new alias entry for the port number (use an "*" wildcard character for the host name). See Configuring virtual hosts for information.

4. Click Finish, and save the changes.

   - The new cluster topology can be viewed from the Servers > Cluster Topology view. If using WebSphere Application Server Version 7.0, the new cluster topology can be viewed from the Servers > Clusters > Cluster Topology view.
   - The Servers > Application Servers view will list the new server cluster members. If using WebSphere Application Server Version 7.0, the Servers > Server Types > WebSphere application servers view will list the new server cluster members.

5. Perform the following steps to enable cache replication:
   A. From the deployment manager Administrative Console, navigate to Servers > Application servers and then click the new vertical cluster member(s). If using WebSphere Application Server Version 7.0, navigate to Servers > Server Types > WebSphere application servers and then click the new vertical cluster member(s).
   B. Click Dynamic cache service under Container services.
   C. Change Cache size to 3000 entries.
   D. Check the Enable cache replication check box.
   E. Select NOT_SHARED from the Replication type drop-down menu.
   F. Click OK.
   G. Click Save to save your changes to the master configuration.

6. Perform the following steps on each vertical cluster member to clean up the server-scoped resources, caches, and resource providers:
   A. Run the ./ConfigEngine.sh cluster-node-config-vertical-cluster-setup -DServerName=unique vertical cluster servername -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, where unique vertical cluster servername is the name you specified when you created the cluster member.
   B. Restart the vertical cluster member referenced in the cluster-node-config-vertical-cluster-setup task.
7. Save your changes and resynchronize the nodes.
   A. In the administrative console for the deployment manager, click **Save** on the task bar, and save your administrative configuration.
   B. Select **System Administration > Nodes**, select the node from the list, and click **Full Resynchronize**.

8. Regenerate the Web server plug-in.
   A. Regenerate the Web server plug-in using the deployment manager administrative console.
   B. If you are using a remote Web server, copy the updated plug-in configuration file (**plugin-cfg.xml**) to the Web server's plug-in configuration directory.

9. Stop and start the Web server.

**Parent topic:** Choosing the type of vertical cluster to create on AIX
Adding vertical cluster members to a dynamic cluster on AIX

You can add vertical cluster members to share the workload demands of your cluster across multiple members running on the same physical machine.

Perform the following steps to add a vertical cluster member to your clustered environment:

1. Open a browser and enter http://DM01:9060/ibm/console in the address bar to access the administrative console on the deployment manager, where DM01 is the deployment manager node or host name. The port number might differ based on your installation.

2. Perform the following steps to allow vertical clusters on your dynamic cluster:
   - Navigate to Servers > Dynamic clusters and select the appropriate dynamic cluster.
   - Select the Allow more than one instance to start on the same node check box under Vertical stacking of Instances on node.
   - Enter a new value in the Number of instances text box to determine the number of vertical cluster members allowed on each node.
   - Click Apply and then click Save to save the changes to the master configuration.

   - The new cluster topology can be viewed from the Servers > Cluster Topology view. If using WebSphere Application Server Version 7.0, the new cluster topology can be viewed from the Servers > Clusters > Cluster Topology view.
   - The Servers > Application Servers view will list the new server cluster members. If using WebSphere Application Server Version 7.0, the Servers > Server Types > WebSphere application servers view will list the new server cluster members.

   Important: You must update the virtual host entries for the new port created when adding a cluster member. You can do this by updating the default_host virtual host in the administrative console and adding a new alias entry for the port number (use an "*" wildcard character for the host name). See Configuring virtual hosts for information.

3. Perform the following steps to enable cache replication:
   - From the deployment manager Administrative Console, navigate to Servers > Application servers and then click the new vertical cluster member(s). If using WebSphere Application Server Version 7.0, navigate to Servers > Server Types > WebSphere application servers and then click the new vertical cluster member(s).
   - Click Dynamic cache service under Container services.
   - Change Cache size to 3000 entries.
   - Check the Enable cache replication check box.
   - Select NOT_SHARED from the Replication type drop-down menu.
   - Click OK.
   - Click Save to save your changes to the master configuration.

4. Perform the following steps on each vertical cluster member to clean up the server-scoped resources, caches, and resource providers:
   - Run the ./ConfigEngine.sh cluster-node-config-vertical-cluster-setup -DServerName=unique vertical cluster servername -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, where unique vertical cluster servername is the name you specified when you created the cluster member.
   - Restart the vertical cluster member referenced in the cluster-node-config-vertical-cluster-setup task.

5. Save your changes and resynchronize the nodes.
   - In the administrative console for the deployment manager, click Save on the task bar, and save your administrative configuration.
B. Select **System Administration > Nodes**, select the node from the list, and click **Full Resynchronize**.

6. Stop and start the Web server.

**Parent topic:** Choosing the type of vertical cluster to create on AIX
Tune your servers

Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

- Prerequisites
  - Preparing prerequisite and corequisite software on AIX
  - Preparing your AIX operating system
  - Preparing the primary node on AIX
  - Choosing the type of cluster to create on AIX

Base Portal Tuning Scenarios

Tuning a WebSphere Portal environment involves tuning and configuring the various systems and components of the environment. The tuning guide provides general concepts and details specifics configuration instructions. Instructions are included for:

- Configuring the application server and the resources defined for that application server
- Determining the cloning strategy for expanding or extending the environment
- Tuning the database(s) and database server
- Tuning the directory server and its database
- Tuning the Web server
- Tuning the operating system and network
- Tuning the WebSphere Portal services
Configuring search in a cluster on AIX

IBM® WebSphere® Portal provides two distinct search capabilities. You can utilize both types of search capabilities in a clustered environment.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on AIX
  - Preparing your AIX operating system
  - Preparing the primary node on AIX
  - Choosing the type of cluster to create on AIX
  - Tune your servers

- **Configuring Portal Search in a cluster on AIX**
  To support Portal Search in a clustered environment, you must install and configure search for remote search service on an IBM WebSphere Application Server node that is not part of the IBM WebSphere Portal cluster.

- **Configuring JCR search in a cluster on AIX**
  To enable search in a cluster for content stored in the JCR database, you must configure each server in the cluster to access a shared directory. JCR-based content includes content created with Web Content Management or Personalization.

**Parent topic:** Setting up a cluster on AIX

**Previous topic:** Tune your servers

**Next topic:** Setting up multiple clusters on AIX
Configuring Portal Search in a cluster on AIX

To support Portal Search in a clustered environment, you must install and configure search for remote search service on an IBM® WebSphere® Application Server node that is not part of the IBM WebSphere Portal cluster.

To install and configure the search service remotely, perform the following tasks:

1. Install and configure the search service to work remotely, that is, on a remote WebSphere Application Server node which is not part of the portal cluster. You can provide the remote search service either as an EJB or as a Web service via SOAP. Deploy the appropriate EJB or SOAP EAR file on the remote WebSphere Application Server node. For details about how to do this, refer to the WebSphere Application Server documentation.

2. Configure the search portlets for remote search service so that they access the remote server accordingly.

Notes:

1. If you have configured a remote search service for a portal cluster, you need to configure the default location for search collections to a directory on the remote server that has write access.

2. The portal site default search collection is created only once at the first time when an administrator selects the search administration portlet Manage Search. If this occurred before you configure the portlet for remote search, then the default portal site search collection is only available on the primary node of the cluster, but not on the remote server. In this case you need to recreate the portal site collection to make it available for search on all nodes of the cluster.

Parent topic: Configuring search in a cluster on AIX

Related concepts
Planning and preparing for Portal Search
Using remote search service

Related tasks
Configuring Portal Search for remote search service
Configuring the Search and Browse portlet for remote search service
Configuring the default location for search collections
Creating or resetting the portal site collection
Configuring the primary node to communicate with the deployment manager on AIX
Configuring the primary node to communicate with the deployment manager on HP-UX
Configuring a remote search service
Configuring JCR search in a cluster on AIX

To enable search in a cluster for content stored in the JCR database, you must configure each server in the cluster to access a shared directory. JCR-based content includes content created with Web Content Management or Personalization. Create a shared directory called `jcr/search` on a server in the network and ensure that each node in the cluster and the Deployment Manager has network access to the directory.

Note: If you are creating content in a clustered environment using the authoring portlet provided with Web Content Management, additional configuration steps are required to enable content created by these content features to be searchable in a cluster.

Perform the following steps on each server in the cluster to configure Search in a clustered environment:

1. Edit the `icm.properties` file, located in the `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm` directory.
2. Change the value of the `jcr.textsearch.indexdirectory` property to the shared directory; for example, `jcr.textsearch.indexdirectory=\\your_server\your_share\jcr\search`. You can specify the shared directory value in one of the following formats: Table 1. The format for the shared directory.

<table>
<thead>
<tr>
<th>Format</th>
<th>Shared directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal Naming Convention (UNC) format</td>
<td><code>\\your_server\your_share\jcr\search</code></td>
</tr>
<tr>
<td></td>
<td><code>\\hostname.example.com\share\jcr\search</code></td>
</tr>
<tr>
<td>Mounted resource format (with forward slashes)</td>
<td><code>/your_share/jcr/search</code></td>
</tr>
<tr>
<td></td>
<td><code>/mnt/jcr/search</code></td>
</tr>
<tr>
<td>Important:</td>
<td>This format requires that you mount the shared directory to the local server (for example, through a mapped network drive or a mounted directory). When using the mounted resource format, always use forward slashes instead of back slashes, regardless of the native operating system path format.</td>
</tr>
</tbody>
</table>

3. Required: Perform the following steps to delete the default search collections from the Manage Search portlet:

A. Log on to WebSphere Portal as an administrator.
B. Click Administration > Search Administration > Manage Search.
C. Click Search Collections.
D. Click the Delete Collection icon for the Portal Content search collection.
E. Click OK.
F. Restart the WebSphere Portal server.
G. Go to the Manage Search portlet and confirm that the Portal Content search collection was deleted.

Parent topic: Configuring search in a cluster on AIX
Setting up multiple clusters on AIX

The majority of the steps to build another cluster are the same as when building the first cluster in the same cell, with a few exceptions. Basically, the new profile will be designated as the primary profile, using IBM® WebSphere® Portal clustering terminology, and will then be used as the basis for the new cluster definition. This duplicates the build process of the first cluster in the cell. During the federation process, if any applications on this new node already exist in the cell (because they are in use by the first cluster), then Deployment Manager will not allow them to be added. After federation, the applications that already exist in the cell are not mapped to the WebSphere_Portal server on the newly federated node, and thus the existing applications must be remapped to this newly federated server to restore its application list. Therefore, depending on the configuration of the new profile, there will likely be some combination of applications shared with the other existing cluster, and some applications unique to this new profile.

Before attempting alternative approaches for building multiple portal-based clusters within a single cell, please contact IBM.

- Prerequisites
  - Preparing prerequisite and corequisite software on AIX
  - Preparing your AIX operating system
  - Preparing the primary node on AIX
  - Choosing the type of cluster to create on AIX
  - Tune your servers

1. Installing multiple clusters in a single cell on AIX

   Create a new, independent IBM WebSphere Portal cluster in a cell where a WebSphere Portal cluster already exists.

2. Routing requests across clusters on AIX

   The HTTP Server plug-in that comes with IBM WebSphere Application Server is typically used to balance requests for applications across members of the cluster. While an application can be mapped to more than one cluster, automatic plug-in generation does not provide routing or balancing traffic for the same application across multiple clusters. Multiple cluster environments with shared applications therefore cannot rely solely on WebSphere Application Server automatic plug-in generation to be able to route requests using a web server. One option in this scenario is developing a customized method for defining and maintaining the plug-in configuration file used by the Web server to provide for the required routing of user requests.

Parent topic: Setting up a cluster on AIX

Previous topic: Configuring search in a cluster on AIX

Next topic: Sharing database domains between clusters on AIX
Installing multiple clusters in a single cell on AIX

Create a new, independent IBM® WebSphere® Portal cluster in a cell where a WebSphere Portal cluster already exists. In the following steps, Cluster A will be used to describe the existing cluster. Portal B will be used to describe the new server profile that will be the basis for the new cluster definition, Cluster B. Perform the following steps to install multiple clusters in a single cell:

1. Upgrade Cluster A, including the Deployment Manager node, to the current, supported hardware and software levels and to the current version of IBM WebSphere Portal.

2. Install and configure Portal B; see the "Preparing the primary node" topic for the appropriate operating system for details. **Important:** Maintain the same number of data sources with identical names to the Cluster A data sources so that data source bindings in the applications can be resolved on every cluster in which they run. If implementing database sharing across the clusters, the above statement refers to both the shared and non-shared domains; all domains should use the same names.

3. Optional: Using the same database user ID and password for each identically named domain/data source will allow the existing JAAS Authentication Aliases to be functional. If unique database user ID and password are required, additional manual configuration is needed to create new JAAS Authentication Aliases for each data source and map these accordingly. On the primary node of Cluster A, run the ./ConfigEngine.sh create-alias-multiple-cluster -DauthDomainList=release,jcr -DWasPassword=dmgr_password task from the wp_profile_root/ConfigEngine directory to create the new JAAS Authentication Aliases. where authDomainList is set to a list of domains which use unique database user ID and passwords and those domain properties are set correctly in the wkplc_comp.properties file, including user ID and password.

4. Optional: If necessary, upgrade Portal B to the current cumulative fix.

5. Run the ./ConfigEngine.sh mapped-app-list-create -DWasPassword=password task from the wp_profile_root/ConfigEngine directory to build an inventory list of Portal B enterprise applications and portlets.

6. Stop the server1 and WebSphere_Portal servers on Portal B and ensure that the following parameters are set correctly in the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory:
   - Set WasSoapPort to the deployment manager's port.
   - Set WasRemoteHostName to the full host name of the deployment manager.

7. If the Deployment Manager is configured to use a stand-alone LDAP user registry, update the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory, on the primary node with the stand-alone LDAP user registry property values from the Deployment Manager. You can find these settings under the VMM Stand-alone LDAP configuration heading.

   **Important:** Ensure that you set WasUserId and WasPassword to the Deployment Manager user ID and password.

8. Run the ./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=password task, from the wp_profile_root/ConfigEngine of the primary node.

9. Run the ./ConfigEngine.sh map-apps-to-server -DWasPassword=password task to determine which applications from the inventory list are no longer mapped to Portal B. The task uses the application profiles already in the cell to restore the mappings. Wait 30 minutes after running this task to allow all EAR files to expand before proceeding to the next step.

10. Perform the following steps to federate Portal B into the deployment manager cell:
    A. Ensure that all database parameters are correctly set, including passwords, in the wkplc_comp.properties and wkplc_dbtype.properties files.
B. Run the `./ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=dmgr_password` task.

C. After running the `cluster-node-config-post-federation` task, wait at least 30 minutes to allow all EAR files to expand.

D. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell’s user registry. Run the `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=waspassword -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task, from the `wp_profile_root/ConfigEngine` directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

Important: If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user` task until after the `cluster-node-config-cluster-setup` (static cluster) or `cluster-node-config-dynamic-cluster-setup` (dynamic cluster) task completes. After running the `wp-change-portal-admin-user` task, start or restart the WebSphere_Portal server to use the updated administrator user ID.

**Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the `wp-change-portal-admin-user` task. `-DnewAdminPw` is an optional parameter to update the Administrative password in the `wkplc.properties` file if required.

E. From the administrative console, click System Administration > Node Agents.

F. Check the box next to the required node agent and then click Restart.

G. Stop and restart the deployment manager.

H. Stop and restart the WebSphere_Portal server on Portal B

11. Restart the WebSphere_Portal server on Cluster A. Verify that Cluster A is functionally intact by spot checking pages and portlets and then verify that Portal B is functionally intact by spot checking pages and portlets that you deployed into Portal B before it was federated. Any discrepancies or errors should be corrected now before continuing. **Note:** If Portal B is using a non-default Portal server administrative ID, not wpsadmin, the server will not be functional until the cluster configuration is complete and the Portal administrative ID has been configured to match the Cells security settings.

12. Choose one of the following options to define a cluster using Portal B as the basis:
Perform the following steps to define a static cluster using Portal B as the basis:

1. Run the `./ConfigEngine.sh cluster-node-config-cluster-setup -DWasPassword=dmgr_password` task.
2. Configure the cluster to use an external Web server to take advantage of features such as workload management. Choose one of the following options:
   - Configuring a Web server and an application server on separate machines (remote)
   - Configuring a Web server and an application server profile on the same machine
   - Configuring a Web server and a deployment manager profile on the same machine

   **Note:** Start with the step about launching the Plug-ins installation wizard.
3. Perform the following steps to access the Web Content Management content through an external Web server:
   - Log on to the deployment manager administrative console.
   - Select `Environment > WebSphere Variables`.
   - From the `Scope` drop-down menu, select the `Node=nodename, Server=servername` option to narrow the scope of the listed variables, where `Node=nodename` is the node that contains the application server.
   - Update the `WCM_HOST` variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.
   - Update the `WCM_PORT` variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.
   - Save your changes and then restart the deployment manager, the node agent(s), server1, and the WebSphere_Portal servers.
   - Change the task list settings to point to the cluster instead of a single server if using Portal business process integration. See one of the following sections in the Information Center for information:
     - Cross cell: Adding a BPI-enabled portal server to a managed cell in a cross-cell setup
     - Single cell: Adding a BPI-enabled portal server to a managed cell in a single-cell setup
Perform the following steps to define a dynamic cluster using Portal B as the basis:

1. Log on to the deployment manager administrative console.
2. Perform the following steps to create a node group:
   - Click **New**.
   - Type the node group **Name**.
   - Type any information about the node group in the **Description** text box.
   - Click **OK**.
   - Click the **Save** link to save your changes to the master configuration.
3. Perform the following steps to add members to the node group:
   - Click **System administration > Node groups**.
   - Click on the name of the node group that you want to add members to.
   - Click **Node group members** under **Additional Properties**.
   - Click **Add**.
   - Select the primary node and then click **Add**.
   - Click the **Save** link to save your changes to the master configuration.
4. Perform the following steps to create a dynamic cluster in the node group:
   - Click **Servers > Dynamic clusters**.
   - Click **New**.
   - Select WebSphere Application Server from the **Server Type** pull-down menu and then click **Next**.
   - Type the cluster name in the **Dynamic cluster name** text box and then click **Next**.
   - Type the same value that you provided for the **ClusterName** parameter in the **wkplc.properties** file of your primary node.
   - Remove all default membership policies and then click **Subexpression builder**.
   - Enter the following information in the Subexpression builder window:
     - Select **Logical operator** pull-down menu.
     - Select **Nodegroup** from the **Select operand** pull-down menu.
     - Select **Equals (=)** from the **Operator** pull-down menu.
     - Type the nodegroup name you created in the previous step in the **Value** text box.
   - Click **Generate subexpression**.
   - Click **Append**.
   - Click **Preview membership** to verify that all nodes included in the nodegroup display and then click **Next**.
   - Click the **Create the cluster member using an existing server as a template** radio button and then select the WebSphere Portal server for the primary node from the pull-down menu.
   - Click **Next**.
   - Specify the dynamic cluster properties and then click **Next**.
   - Review the summary page to verify your actions and then click **Finish**.

Define or verify the following parameters in the **wkplc.properties** file, located in the **wp_profile_root/ConfigEngine/properties** directory:

- **ClusterName** to the name of the new dynamic cluster.
- **CellName** is set to the deployment manager cell.
- **NodeName** is set to the local WebSphere Portal node.
- **ServerName** to the server that will be used for the dynamic cluster member on this node.
- **PrimaryNode** is set to true.

Run the 

```
./ConfigEngine.sh cluster-node-config-dynamic-cluster-setup -DWasPassword=dmgr_password
```
task from

the **wp_profile_root/ConfigEngine directory** to create the dynamic cluster.
13. Install any additional nodes to the cell to support additional cluster members for **Cluster B** identically to the primary node, and then federate as them as secondary nodes and define as cluster members on these nodes. For information about adding additional nodes navigate to **Installing WebSphere Portal > Setting up WebSphere Portal > Setting up a clustered production environment**. Select the appropriate operating system and navigate to **Preparing additional nodes**. You can then choose to add an additional static or dynamic cluster and/or a vertical static or dynamic cluster.

14. Restart the server1 and WebSphere_Portal servers on **Cluster A** and **Cluster B**.

15. If you entered passwords in any of the properties files while creating your cluster, you should remove them for security purposes. See "Deleting passwords from properties files" under Related tasks for information.

Installation of **Cluster B** is complete. It is now an independent cluster from **Cluster A**, which means that **Cluster B** can have its own configuration, set of end-user portlets, and target community. Any applications that are common between **Cluster A** and **Cluster B** are most likely infrastructure or related to administration, and special care needs to be taken to preserve their commonality between clusters and correct maintenance levels.

**Parent topic:** Setting up multiple clusters on AIX

**Next topic:** Routing requests across clusters on AIX
Related tasks
Deleting passwords from properties files
Adding a BPI-enabled portal server to a managed cell in a cross-cell setup
Adding a BPI-enabled portal server to a managed cell in a single-cell setup
Routing requests across clusters on AIX

The HTTP Server plug-in that comes with IBM® WebSphere® Application Server is typically used to balance requests for applications across members of the cluster. While an application can be mapped to more than one cluster, automatic plug-in generation does not provide routing or balancing traffic for the same application across multiple clusters. Multiple cluster environments with shared applications therefore cannot rely solely on WebSphere Application Server automatic plug-in generation to be able to route requests using a web server. One option in this scenario is developing a customized method for defining and maintaining the plug-in configuration file used by the Web server to provide for the required routing of user requests.

An important consideration in a multiple cluster environment is ensuring that all subsequent HTTP requests for an end user are routed to the same cluster that processed the first HTTP request. The WebSphere Portal login processing depends upon preserving this cluster affinity during this initial time until the user has successfully logged in and session cookies maintain affinity. In order to guarantee that affinity is preserved during login, set the Navigator Service public.session parameter to a value of true. Refer to “Portal Configuration Services” for information on how to configure this parameter.

Parent topic: Setting up multiple clusters on AIX
Previous topic: Installing multiple clusters in a single cell on AIX
Sharing database domains between clusters on AIX

To help provide redundancy and failover support in production environments composed of multiple clusters in a single cell and multiple clusters in different cells, you can share database domains between those clusters. IBM® WebSphere® Portal data is organized into different database domains, with different availability requirements depending on how the production environment is set up. When multiple lines of production are involved and each line of production is implemented as a cluster of servers, sharing database domains ensures that the data is automatically synchronized across the lines of production.

**Prerequisites**
- Preparing prerequisite and corequisite software on AIX
- Preparing your AIX operating system
- Preparing the primary node on AIX
- Choosing the type of cluster to create on AIX
- Tune your servers

**Important:** JCR domains and release domains cannot be shared between different Web Content Management clusters or servers. Each distinct cluster or server in your Web Content Management system must use a separate JCR or release domain. For example:

<table>
<thead>
<tr>
<th>Development Server</th>
<th>Authoring Cluster</th>
<th>Staging Server</th>
<th>Delivery Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCR Domain 1</td>
<td>JCR Domain 2</td>
<td>JCR Domain 3</td>
<td>JCR Domain 4</td>
</tr>
<tr>
<td>Release Domain 1</td>
<td>Release Domain 2</td>
<td>Release Domain 3</td>
<td>Release Domain 4</td>
</tr>
</tbody>
</table>

Perform the following steps to share database domains when setting up an environment with multiple clusters:

1. Set up the first cluster (referred to as Cluster A in these instructions).
2. Determine which database domains you want to share with any other clusters in the environment.
3. Install the primary node of the next cluster (Cluster B), and perform the following steps to configure the node to use the shared database domains.
   A. Perform a partial database transfer of the database domains that you are not sharing. For example, if you are sharing only the Customization and Community domains, you would transfer the remaining domains to the database you are using for Cluster B.
   B. Reconfigure the shared database domains on the node to connect to the shared database domains you are using for Cluster A. For example, to connect to the Customization and Community domains for Cluster A, you would connect to those domains from Cluster B.
4. Continue setting up the primary node as described in the cluster instructions.
5. Install the secondary node of Cluster B, and perform the following steps to configure the node to use the shared database domains.
   A. For those database domains that you are not sharing between clusters, reconfigure the domains to connect to the database domains you are using for Cluster B. As in the example for the primary node, if you are sharing only the Customization and Community domains, reconfigure the remaining domains on the secondary node to use the domains of the primary node for Cluster B.
   B. Reconfigure the shared database domains on the node to connect to the shared database domains you are using for Cluster A. For example, to connect to the Customization and Community domains for Cluster A, you would connect to
those domains from Cluster B.

6. Continue setting up the secondary node as described in the cluster instructions.

**Parent topic:** Setting up a cluster on AIX  
**Previous topic:** Setting up multiple clusters on AIX

**Related tasks**  
Connecting to existing database domains
Setting up a cluster on HP-UX

In a production environment, you can install and configure IBM® WebSphere® Portal on a single (primary) node and then add additional nodes in the future as your company's needs grow. You can also deploy a high availability production environment by creating and configuring clustered deployments of multiple servers running WebSphere Portal managed by IBM WebSphere Application Server Network Deployment.

**Prerequisites**
- Technotes for installation and configuration issues

Perform the following tasks to set up your production environment on HP-UX:

1. **Preparing prerequisite and corequisite software on HP-UX**
   Before creating your high availability production environment, you must prepare your database, your user registry, your deployment manager, and your remote Web server.

2. **Preparing your HP-UX operating system**
   View information on setting up your operating system for IBM WebSphere Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

3. **Preparing the primary node on HP-UX**
   Before creating your high availability environment, you must install IBM WebSphere Portal on your primary node and then configure your database and your network deployment manager.

4. **Choosing the type of cluster to create on HP-UX**
   If you installed a IBM WebSphere Application Server Network Deployment, create a static cluster to handle failover requests. If you installed a IBM WebSphere Virtual Enterprise, create a dynamic cluster to balance member workloads.

5. **Preparing a remote Web server when portal is installed on HP-UX**
   Install and configure the Web server plug-in, provided by IBM WebSphere Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

6. **Configuring WebSphere Portal to use a user registry on HP-UX in a clustered environment**
   Configure user registry security on IBM WebSphere Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

7. **Preparing additional nodes on HP-UX**
   After installing and configuring your primary node, you can create your secondary nodes. You must install IBM WebSphere Portal on each node and then configure the node to access the database and user registry before adding it to the cluster.

8. **Tune your servers**
   Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

9. **Configuring search in a cluster on HP-UX**
   IBM WebSphere Portal provides two distinct search capabilities. You can utilize both types of search capabilities in a clustered environment.
10. Setting up multiple clusters on HP-UX

The majority of the steps to build another cluster are the same as when building the first cluster in the same cell, with a few exceptions. Basically, the new profile will be designated as the primary profile, using IBM WebSphere Portal clustering terminology, and will then be used as the basis for the new cluster definition. This duplicates the build process of the first cluster in the cell. During the federation process, if any applications on this new node already exist in the cell (because they are in use by the first cluster), then Deployment Manager will not allow them to be added. After federation, the applications that already exist in the cell are not mapped to the WebSphere_Portal server on the newly federated node, and thus the existing applications must be remapped to this newly federated server to restore its application list. Therefore, depending on the configuration of the new profile, there will likely be some combination of applications shared with the other existing cluster, and some applications unique to this new profile.

11. Sharing database domains between clusters on HP-UX

To help provide redundancy and failover support in production environments composed of multiple clusters in a single cell and multiple clusters in different cells, you can share database domains between those clusters. IBM WebSphere Portal data is organized into different database domains, with different availability requirements depending on how the production environment is set up. When multiple lines of production are involved and each line of production is implemented as a cluster of servers, sharing database domains ensures that the data is automatically synchronized across the lines of production.

Parent topic: Setting up a cluster
Preparing prerequisite and corequisite software on HP-UX

Before creating your high availability production environment, you must prepare your database, your user registry, your deployment manager, and your remote Web server.

Perform the following tasks to prepare prerequisite and corequisite software:

1. Preparing the WebSphere Application Server Deployment Manager on HP-UX
   IBM WebSphere Portal provides a customized installation package (CIP) that includes IBM WebSphere Application Server Network Deployment and all required maintenance packages. Use the supplied discs to install WebSphere Application Server Network Deployment on a dedicated system.

2. Preparing a remote Web server when portal is installed on HP-UX
   Install and configure the Web server plug-in, provided by IBM WebSphere Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

Parent topic: Setting up a cluster on HP-UX
Next topic: Preparing your HP-UX operating system
Preparing the WebSphere Application Server Deployment Manager on HP-UX

IBM® WebSphere® Portal provides a customized installation package (CIP) that includes IBM WebSphere Application Server Network Deployment and all required maintenance packages. Use the supplied discs to install WebSphere Application Server Network Deployment on a dedicated system.

Perform the following steps to prepare the WebSphere Application Server Deployment Manager:

1. Install WebSphere Application Server Network Deployment or use the CIP to upgrade an existing installation. Run the following command: `cd_root/HP-UX/architecture/ifpackage/WAS/install`, where `cd_root` is the root directory of the disc and `architecture` is the system's processor architecture.

2. Choose one of the following options to create a default deployment manager profile:

   **Important:** While creating the default deployment manager profile, enable administrative security. If you use the Profile Management Tool, check the enable administrative security check box. If you use the `manageprofile` command, add the `-enableAdminSecurity=true` parameter to the command line.

3. Run the following command to start the deployment manager: `./startManager.sh`, from the `dmgr_profile_root/bin` directory.

4. Use the following URL to launch the network deployment administrative console: `http://dmgr_hostname:9060/ibm/console`, where `dmgr_hostname` is the fully qualified host name for the WebSphere Application Server Network Deployment.

5. Log into the deployment manager administrative console.

6. Increase the HTTP connection timeouts for the deployment manager.

   A. Click **System Administration** > Deployment Manager > Web container transport chains.

   B. Increase the timeout values. For the `WCinboundAdmin` and `WCinboundAdminSecure` entries listed in the web container transport chains section, complete the following steps to increase the timeout values:

      1. Click **HTTP Inbound Channel**.
      2. Change the **Read timeout** value to 180.
      3. Change the **Write timeout** value to 180.
      4. Save the configuration changes.

7. Change the timeout request period for the Java Management Extensions (JMX) connector.

   A. Click **System administration** > Deployment Manager > Administration Services > JMX connectors > SOAPConnector > Custom Properties.

   B. Select the `requestTimeout` property, and increase the value from 600 to 6000.

   C. Save the configuration changes.

8. Update the maximum Java heap size used by the deployment manager:

   A. Click **System administration** > Deployment manager > Java and Process Management > Process Definition > Java Virtual Machine.

---

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile Management Tool</td>
<td>See Creating a deployment manager profile for information.</td>
</tr>
<tr>
<td><code>manageprofile</code> commands</td>
<td>See manageprofile commands for information. <strong>Note:</strong> If you have a 64-bit</td>
</tr>
<tr>
<td></td>
<td>environment, only the manageprofiles command is supported when creating</td>
</tr>
<tr>
<td></td>
<td>profiles.</td>
</tr>
</tbody>
</table>

---

**Operating systems:** AIX, HP-UX, i5/OS, Linux, Solaris, Windows
B. Update the value in the **Maximum Heap Size** field. For information about appropriate heap sizes see the documentation for your operating system in the Performance Guides located on the WebSphere Portal and Web Content Management *Product Documentation* page. **Note:** If using a 32-bit operating system, you will need to set the heap size to a lower size than a 64-bit operating system.

C. Click **OK** and then save your changes.

9. Depending on your version of WebSphere Application Server, make the appropriate selection from the navigation:

   **Note:** If security is not enabled on your deployment manager, see *Enabling security* for information before performing this step.

   - For **WebSphere Application Server Version 6.1**: Click **Security > Secure administration, applications and infrastructure** and select **Enable Application Security**. Then save the configuration changes.

   - For **WebSphere Application Server Version 7.0**: Click **Security > Global security** and select **Enable Application Security**. Then save the configuration changes.

10. Verify that the WebSphere Portal administrative users and administrative group exist in the Deployment Manager cell's user registry. Perform the following steps if you need to create the administrative users and group:

   A. Click **Users and Groups > Manage Users**.

   B. Click **Create**.

   C. Type the information for the WebSphere Portal administrative users; for example *wpsadmin* and *wpsbind*, and then click **Create**.

   D. Click **Users and Groups > Manage Groups**.

   E. Click **Create**.

   F. Type *wpsadmins* as the name of the WebSphere Portal administrative group and then click **Create**.

   G. Click the group you just created; for example *wpsadmins*.

   H. Click the **Members** tab.

   I. Click **Add Users**.

   J. Search for the users.

   K. Select the users you want to add to the group.

   L. Click **Add** to add the users to the group.

   M. Click **Close** when you are done adding users to the group.

   N. Log out of the administrative console.

11. Change the timeout request period for the Simple Object Access Protocol (SOAP) client. Edit the *soap.client.props* file, located in the *Dmgr_profile/properties* directory: Change the line to: `com.ibm.SOAP.requestTimeout=6000`.

12. Run the following tasks to stop and restart the deployment manager:

   A. `. /stopManager.sh -username admin_userid -password admin_password`, from the *dmgr_profile_root/bin* directory

   B. `. /startManager.sh`, from the *dmgr_profile_root/bin* directory

**Parent topic:** Preparing prerequisite and corequisite software on HP-UX

**Next topic:** Preparing a remote Web server when portal is installed on HP-UX
Preparing a remote Web server when portal is installed on HP-UX

Install and configure the Web server plug-in, provided by IBM® WebSphere® Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

- **Prerequisites**
  - Preparing the WebSphere Application Server Deployment Manager on HP-UX
  - Preparing prerequisite and corequisite software on HP-UX
  - Preparing your HP-UX operating system
  - Preparing the primary node on HP-UX
  - Choosing the type of cluster to create on HP-UX

Perform the following steps to install and configure your Web server:

1. Install and configure the Web server; refer to the Web server documentation for information.

2. If using IBM Lotus® Domino®, edit the `NOTES.INI` file on the Web server. Set the `HTTPEnableConnectorHeaders` parameter to 1.

3. If using IBM HTTP Server or Apache Server, edit the `httpd.conf` file on the Web server. Set the `AllowEncodedSlashes` directive to `On`; the directive should be added at the root level as a global directive.

4. Stop the Web server.

5. Install and configure the Web server plug-in on the system where the Web server is located using the Plug-ins installation wizard provided with WebSphere Application Server. Refer to the Web server topology diagram and roadmap for information. **Note:** The WebSphere Application Server Web server instructions are for the default profile. If not using the default profile, refer to Configuring a Web server for a non-default profile.

   **If using WebDAV:** After successfully installing the Web server plug-in, locate and open your `plugin-cfg.xml` file and set `AcceptAllContent` to `true`.

   **Important:** Depending on how you use the Web server, you may need to adjust the `ServerIOTimeout` value, which defines how long the plug-in should wait for a response from the application. The recommended minimum value is 60 but you may need to adjust this value higher if you are retrieving data from a database. To update this value, locate and open your `plugin-cfg.xml` file and set `ServerIOTimeout` to a value that is appropriate for your business needs. For additional information, see Common questions about the Web server plug-in.

6. If you are using Sun Web Server Version 7 update 8, read Technote 1448262 and perform the steps to resolve the HTTP 408/409 error.

7. Start the Web server.
Preparing your HP-UX operating system

View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

- Prerequisites
  - Preparing prerequisite and corequisite software on HP-UX

Configuring HP-UX kernel settings before installing

To set kernel parameters, perform the following steps:

1. Log into the host machine with superuser (root) privileges.
2. Choose one of the following options to determine the physical memory, which you must know to avoid setting certain kernel parameters above the physical capacity:
   - HP-UX 11i Version 3 and later: Use the /usr/sbin/dmesg command.
   - HP-UX 11i Version 2 and earlier: Perform the following steps:
     A. Start the HP-UX System Administration Manager (SAM) utility with the /usr/sbin/sam command.
     B. Select Performance Monitors > System Properties > Memory.
     C. View the value for Physical Memory and click OK.
     D. Exit from the SAM utility.
3. Set the maxfiles and maxfiles_lim parameters to at least 4096. (The table below recommends 8000 and 8196, respectively. You must first edit the /usr/conf/master.d/core-hpux file, to allow the SAM utility to set values greater than 2048: Note; If using HP-UX 11i Version 2, the /usr/conf/master.d/core-hpux file does not exist; instead, use the SAM utility to set the new kernel parameter values. See Reconfiguring the Kernel (HP-UX 11i Version 2) for details.
   A. Open the file /usr/conf/master.d/core-hpux in a text editor.
   B. Change the line, "*range maxfiles<=2048" to "*range maxfiles<=60000"
   C. Change the line, "*range maxfiles_lim<=2048" to "*range maxfiles_lim<=60000"
   D. Save and close the file. Old values might be stored in the /var/sam/boot.config file. Force the SAM utility to create a new boot.config file:
      1. Move the existing version of the /var/sam/boot.config file to another location, such as the /tmp directory.
      2. Start the SAM utility.
      3. Select Kernel Configuration > Configurable Parameters. When the Kernel Configuration window opens, a new boot.config file exists. Alternatively, rebuild the boot.config file with this command: #
         /usr/sam/lbin/getkinfo -b
4. Web Content Management only: Use the ulimit -f command to set the maximum size of files that can be created to be at least the size of the largest file you would need to upload to the content server. The command ulimit -f -1 removes any limit on file size.
5. Set new kernel parameter values:
   A. Start the SAM utility.
   B. Select Kernel Configuration > Configurable Parameters.
   C. For each of the parameters in the following table, perform this procedure:
      1. Highlight the parameter to change.
      2. Select Actions > Modify Configurable Parameter.
      3. Type the new value in the Formula/Value field.
4. Click OK.

**Note:** When WebSphere Portal and DB2 are installed on the same machine, some kernel values are higher than those shown in the table below. If you are using this configuration, refer to the following site for more information:


Typical kernel settings for running WebSphere Portal appear in the following table: *Table 1. The typical kernel settings for running WebSphere Portal*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbc_max_pct</td>
<td>25</td>
</tr>
<tr>
<td>maxdtsiz</td>
<td>805306358</td>
</tr>
<tr>
<td>maxdtsiz</td>
<td>2048000000 (when WebSphere Application Server Base and Network Deployment products are on the same machine)</td>
</tr>
<tr>
<td>maxfiles_tip</td>
<td>8196 (Change this one before maxfiles.)</td>
</tr>
<tr>
<td>maxfiles</td>
<td>8000</td>
</tr>
<tr>
<td>maxssiz</td>
<td>8388608</td>
</tr>
<tr>
<td>maxswapchunks (valid for HP-UX 11iv1 only)</td>
<td>8192</td>
</tr>
<tr>
<td>max_thread_proc</td>
<td>3000</td>
</tr>
<tr>
<td>maxuprc</td>
<td>512</td>
</tr>
<tr>
<td>maxusers (valid for HP-UX 11iv1 only)</td>
<td>512</td>
</tr>
<tr>
<td>msgmap</td>
<td>2048</td>
</tr>
<tr>
<td>msgmax</td>
<td>65535</td>
</tr>
<tr>
<td>msgmax</td>
<td>131070 (when WebSphere Application Server Base and Network Deployment products are on the same machine)</td>
</tr>
<tr>
<td>msgmnb</td>
<td>65535</td>
</tr>
<tr>
<td>msgmnb</td>
<td>131070 (when WebSphere Application Server Base and Network Deployment products are on the same machine)</td>
</tr>
<tr>
<td>msgmrni</td>
<td>50</td>
</tr>
<tr>
<td>msgseg</td>
<td>32767</td>
</tr>
<tr>
<td>msgsssz</td>
<td>96</td>
</tr>
<tr>
<td>msgtql</td>
<td>2046</td>
</tr>
<tr>
<td>nfile</td>
<td>58145</td>
</tr>
<tr>
<td>nflags</td>
<td>3000</td>
</tr>
<tr>
<td>ninode</td>
<td>60000</td>
</tr>
<tr>
<td>nkthread</td>
<td>7219</td>
</tr>
<tr>
<td>nproc</td>
<td>4116</td>
</tr>
<tr>
<td>npt</td>
<td>2024</td>
</tr>
<tr>
<td>nstrpty</td>
<td>1024</td>
</tr>
<tr>
<td>nstrtel</td>
<td>60</td>
</tr>
<tr>
<td>sema (valid for HP-UX 11iv1 only)</td>
<td>1 (with Embedded Messaging)</td>
</tr>
<tr>
<td>semaen</td>
<td>16384 (with Embedded Messaging)</td>
</tr>
<tr>
<td>semmap (valid for HP-UX 11iv1 only)</td>
<td>514</td>
</tr>
<tr>
<td>semmni</td>
<td>2048</td>
</tr>
<tr>
<td>semmns</td>
<td>16384 (with Embedded Messaging)</td>
</tr>
<tr>
<td>semmnu</td>
<td>1024</td>
</tr>
<tr>
<td>semume</td>
<td>200</td>
</tr>
<tr>
<td>semvmx</td>
<td>32767 (with Embedded Messaging)</td>
</tr>
<tr>
<td>shmem</td>
<td>2147483647</td>
</tr>
<tr>
<td>shmem (valid for HP-UX 11iv1 only)</td>
<td>1 (with Embedded Messaging)</td>
</tr>
</tbody>
</table>
6. Select **Actions > Process New Kernel**.
7. Click **Yes** on the information window to confirm your decision to restart the machine.
8. Follow on-screen instructions to restart your machine and to enable the new settings.

### Table

<table>
<thead>
<tr>
<th>shmmni</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>shmseg</td>
<td>1024</td>
</tr>
<tr>
<td>STRMSGSZ (valid for HP-UX 11iv1 only)</td>
<td>65535</td>
</tr>
</tbody>
</table>

**Parent topic:** Setting up a cluster on HP-UX  
**Previous topic:** Preparing prerequisite and corequisite software on HP-UX  
**Next topic:** Preparing the primary node on HP-UX
Preparing the primary node on HP-UX

Before creating your high availability environment, you must install IBM® WebSphere® Portal on your primary node and then configure your database and your network deployment manager.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on HP-UX
  - Preparing your HP-UX operating system

Perform the following tasks to prepare your primary node:

1. **Installing WebSphere Portal on HP-UX on the primary node**
   IBM WebSphere Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

2. **Configure WebSphere Portal to use a remote database**
   For high availability, using a remote database is preferred. For improved performance databases can be distributed across multiple database servers. Databases should also be configured for failover. Configuring the database for failover is beyond the scope of this documentation. Refer to the database product documentation for the most authoritative guidance about setting up databases for fail over.

3. **Configuring the primary node to communicate with the deployment manager on HP-UX**
   After installing IBM WebSphere Application Server Network Deployment and installing IBM WebSphere Portal on the primary node, you must configure the primary node and the deployment manager to communicate with each other.

4. **Removing search collections on HP-UX**
   If you plan to use search in a cluster, you must configure a remote search server; see Configuring Portal Search in a cluster for information. If you created any search collections, you will need to recreate them on the remote search server. If your search collection has data, export the collection before deleting it and then import the data on the remote server.

**Parent topic:** Setting up a cluster on HP-UX  
**Previous topic:** Preparing your HP-UX operating system  
**Next topic:** Choosing the type of cluster to create on HP-UX
Installing WebSphere Portal on HP-UX on the primary node

IBM® WebSphere® Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

Review the Supported hardware and software requirements for your operating system and ensure that all hardware and software matches the minimum product level before installing WebSphere Portal. Review Installation methods, options, and sources before installing to determine your installation source location and installation method.

Perform the following steps to install WebSphere Portal and create a custom profile for the node that will be added to the cluster. If you already installed WebSphere Portal, you can skip to the last step to create the custom profile.

1. Type `ping yourserver.yourcompany.com` on a command line to verify that your fully qualified host name is properly configured.
2. Type `ping localhost` on a command line to verify that your network is properly configured.
3. If you are installing with an existing WebSphere Application Server instance, ensure it is installed at the supported level and has the following features installed:
   - Application Server
   - Administration
   - Scripted Administration
   - Administrative Console
   - Ant and Deployment Tools
   - Deploy Tool
   - Ant Utilities

   **Restriction:** Installing WebSphere Portal Version 6.1.0.1 or higher on an existing WebSphere Application Server Version 7.0 instance where WebSphere Portal Version 7.0 was installed is unsupported.

   **Important:** Besides ensuring that the existing WebSphere Application Server instance is at the supported level, you will also need to ensure that Apache Derby is installed at the supported level before installing WebSphere Portal. See WebSphere Portal detailed system requirements for supported levels.

4. Optional: Perform the following steps if you want to install WebSphere Portal using a non-root user:

   **Restriction:** Although it is possible to install WebSphere Application Server as a non-root user, there are limitations to this option that can affect WebSphere Portal functionality, which is why you should install WebSphere Application Server as a root user.

   A. Install the current supported version of WebSphere Application Server as a root user.
   B. Open a command prompt and perform the following steps to create a non-root user and to change ownership:

   1. Use the appropriate system commands to create a new group, to create a new user, and to add the new user to the new group.

   2. Run the following tasks to change the rights of the non-root user:

   ```
   chgrp -R group_name/opt/IBM
   chmod -R g+rw /opt/IBM
   chmod -R g+wr /tmp
   ```
chgrp -R group_name /tmp
chmod -R g+wr /var/tmp
chgrp -R group_name /var/tmp

C. Launch the installation program per the next step. During the installation, select the existing WebSphere Application Server using the non-root user and set the installation location to a unique location under the path where non-root permissions were granted.

5. Choose one of the following installation commands based on the installation method you want to use to install WebSphere Portal; see Installation Methods for more information:

**Advance Installation parameters:** To customize your installation, you can add various parameters to your installation commands; for example, you can specify your port information. See Advanced installation parameters for information.

**Restriction:** When defining your installation path, the ONLY supported characters are ASCII letters [A-Z and a-z], numbers [0-9], slashes [/ or \\, depending on your operating system], and the dash [-].

<table>
<thead>
<tr>
<th>Table 1. Installation task options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Installation type</strong></td>
</tr>
<tr>
<td>-----------------------</td>
</tr>
<tr>
<td><strong>Graphical user interface</strong></td>
</tr>
<tr>
<td><strong>Console mode</strong></td>
</tr>
<tr>
<td><strong>Silent install</strong></td>
</tr>
</tbody>
</table>

**Note:** If the installation program does not detect a WebSphere Application Server instance that you know exists, exit the installation program and pass the WebSphere Application Server instance location using the command line; for example, `./install.sh -W was.undetectedWas="/my/WAS/location"`.

**Note:** If the GUI or console mode installation program fails to detect ports for either WebSphere Application Server or WebSphere Portal, a warning message displays and the installer offers another chance to enter the values. If the silent installation fails to detect ports for either WebSphere Application Server or WebSphere Portal, the installer will exit.

6. Verify your installation was successful; access WebSphere Portal using the `http://yourserver:yourport/wps/portal` format. Run the `./ConfigEngine.sh list-server-ports -DWasPassword=passwort` task from the `wp_profile_root/ConfigEngine` directory to generate the `wp_profile_root/ConfigEngine/log/wp_PortMatrix.txt` file that lists WebSphere Portal ports for your installation.

7. Optional: After the WebSphere Portal installation completes successfully, run the `./ConfigEngine.sh configure-express -DPortalAdminPwd=your_password -DWasPassword=your_password` task, from the `wp_profile_root/ConfigEngine` directory, if you want to enable the sample IBM Lotus Web Content Management content, such as internet and intranet sample sites.

**Restriction:** Run the `configure-express` task before configuring your database, user registry, context root, security, etc. If you run any tasks other than the install task, do not run this task.

**Note:** See [http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_ovw.html](http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_ovw.html) for tutorials on how to use the sample content.

The sample content includes:

- Creates a group called contentAuthors; members of this group are given privileges to create content in the sample...
Internet and intranet sites. Navigate to the Administration area and then click Access > Users and Groups.
- Creates two new Web Content Management Libraries: "Internet Web Content 6.1.0" and "Intranet Web Content 6.1.0". Navigate to the Administration area and then click Portal Content > Web Content Libraries.
- Adds a portlet filter and applies the filter to various portlets in the sample Internet and intranet sites. You can see the definition of the filter in the WebSphere Application Server Administration console and examining the custom resources under the Environment area.
- Creates two new theme policies: InternetStyle and IntranetStyle. These styles are applied to sample Internet and intranet sites. Navigate to Theme Customizer and then select the style.
- Creates several portlet clones of the Web Content Management rendering portlet. These portlet clones are used on sample Internet and intranet sites.
- Creates two virtual portals with context roots of wps/portal/intranet and wps/portal/internet. These are the sample Internet and intranet sites. Go to http://yourserver:yourport/wps/portal/internet and http://yourserver:yourport/wps/portal/intranet to access them.
- Creates several sample credential slots, including "Default slot for E-mail", "Default slot for Feeds", "Default slot for Miscellaneous", "Default slot for Web Clipping", and "Default slot for Web Content Management". Navigate to the Administration area and then click Access > Credential Vault > Manage System Vault Slots.
- Modifies the portlet palette to include some new portlets. Click the Portlets link to see the portlet palette.

8. Optional: If you ran the configure-express task, the owner of the items in the Web content libraries containing the Internet and Intranet Site Template content will be listed as uid=xyzadmin,o=defaultWIMFileBasedRealm. To update the owner information for these items to correspond to your portal administrator ID, complete the following steps:
   A. Edit the wp_profile_root /PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties file.
   B. Add the following line to the file:
      
      uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN

      Replace portal_admin_DN with the distinguished name of the portal administrator.
   C. Save your changes and close the file.
   D. Run the ./ConfigEngine.sh action-express-memberfixer -DPortalAdminPwd=password -DWasPassword=password task, located in the wp_profile_root/ConfigEngine directory.

9. Optional: Perform the following steps if you need to change the ports for WebSphere Application Server or WebSphere Portal:

   **Note:** The starting port parameter is required for a successful completion of the modify-ports-by-startport task.

   Once you specify a start port, this port becomes the base for assigning port values. The code increments this value as each port is assigned, which means that the WebSphere Portal ports will be assigned incrementally starting with the port defined with the modify-ports-by-startport task.

   A. Stop the server1 and WebSphere_Portal servers.
   B. Run one of the following commands for each server you need to change:

<table>
<thead>
<tr>
<th>Method</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting port number</td>
<td>./ConfigEngine.sh modify-ports-by-startport -DWasPassword=password -DModifyPortsServer=servername -DStartPort=starting port number</td>
</tr>
</tbody>
</table>
C. Restart the server1 and WebSphere_Portal servers.

10. To prevent Out of Memory errors, perform the following steps to set the **MaxPermSize**: Tip: If **Maximum Heap Size** is set to 2048 M or higher, set the **MaxPermSize** to a quarter of the value entered for the **Maximum Heap Size**; for example, if your **Maximum Heap Size** is 3000 M, set **MaxPermSize** to 750 M. If your **Maximum Heap Size** is less than 2048 M, set **MaxPermSize** to 512 M.

A. Log in to the WebSphere Application Server Administration Console.

B. Click **Servers > Server Types > WebSphere application servers > WebSphere_Portal**.


D. In the **Generic JVM arguments** field, change the **MaxPermSize** value to `-XX:MaxPermSize=numeric value`, where `numeric value` is a quarter of the value entered for the **Maximum Heap Size**.

Important: **MaxPermSize** does not exist in the **Generic JVM arguments** field, add it to the field but do not replace existing information in the **Generic JVM arguments** field with the **MaxPermSize** information.

E. Click **OK** to save your changes.

F. Click **Save** to save your changes to the master configuration.

G. Log out of the WebSphere Application Server Administration Console.

H. Restart the server1 and WebSphere_Portal servers.

**Port file**

**Note:** Sample port files are available on the Setup disc.

```bash
./ConfigEngine.sh modify-ports-by-portsfile -
DWasPassword=password -
DModifyPortsServer=servername -
DPortsFile=full path to ports file
```

The following is an example of the information within a port file although the port values will be different based on your environment:

- `BOOTSTRAP_ADDRESS=10031`
- `SOAP_CONNECTOR_ADDRESS=10031`
- `SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=10032`
- `CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=10025`
- `CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=10036`
- `WC_adminhost=10027`
- `WC_defaulthost=33344`
- `DCS_UNICAST_ADDRESS=10029`
- `WC_adminhost_secure=10039`
- `WC_defaulthost_secure=10035`
- `SIB_ENDPOINT_ADDRESS=10026`
- `SIB_ENDPOINT_SECURE_ADDRESS=10037`
- `SIB_MQ_ENDPOINT_ADDRESS=10030`
- `SIB_MQ_ENDPOINT_SECURE_ADDRESS=10028`
- `ORB_LISTENER_ADDRESS=10034`

**Related concepts**

Installation methods
IBM Support Assistant Lite for WebSphere Portal

**Related reference**

Advanced installation parameters
Configure WebSphere Portal to use a remote database

For high availability, using a remote database is preferred. For improved performance databases can be distributed across multiple database servers. Databases should also be configured for failover. Configuring the database for failover is beyond the scope of this documentation. Refer to the database product documentation for the most authoritative guidance about setting up databases for fail over.

Password considerations when transferring data manually

For security reasons, you should not store passwords in the wkplc.properties and wkplc_comp.properties. Edit each of the properties files prior to running a configuration task, inserting the passwords needed for that task. After the task has run, delete all passwords from each file. For more information, see Deleting passwords from configuration scripts.

Alternatively, you can specify the password on the command line using the following syntax:

- Windows: ConfigEngine.bat task_name -Dpassword_property_key=password_value
- UNIX: ./ConfigEngine.sh task_name -Dpassword_property_key=password_value
- i5/OS: ConfigEngine.sh task_name -Dpassword_property_key=password_value

As with other properties, each password property must have the -D prefix and be set equal to (=) a value. If you have multiple properties in a single command, use a space character between each -Dproperty=value setting.

- Prerequisites
  - Installing WebSphere Portal on HP-UX on the primary node
  - ▶ Technotes for database connectivity issues

- Configuring WebSphere Portal to use DB2
  View information on installing and setting up DB2 to work with WebSphere Portal.

- Configuring WebSphere Portal to use DB2 for z/OS
  View information on installing and setting up DB2 for z/OS to work with WebSphere Portal.

- Configuring WebSphere Portal to use Oracle
  View information on installing and setting up Oracle to work with WebSphere Portal.

- Configuring WebSphere Portal to use Oracle RAC
  View information on installing and setting up Oracle RAC to work with WebSphere Portal.

- Configuring WebSphere Portal to use SQL Server
  View information on installing and setting up SQL Server to work with WebSphere Portal.

- Verifying databases
  After you configure IBM WebSphere Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.
Troubleshooting WebSphere Portal Version 6.1 databases
Configuring WebSphere Portal to use DB2

View information on installing and setting up DB2 to work with WebSphere Portal.

1. **Installing DB2**
   View information on installing DB2 for use with WebSphere Portal.

2. **Create users**
   View the steps to set up users for DB2 to work with WebSphere Portal.

3. **Creating remote databases**
   A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

4. **Optional: Assigning custom table spaces**
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

5. **Modifying database properties**
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

6. **Optional: Configuring JCR collation support**
   View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

7. **Configure WebSphere Portal to use DB2**
   View the steps to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

8. **Configuring DB2 for large file handling in Web Content Management**
   If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

9. **Optional: Changing driver types**
   WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

**Parent topic:** Configure WebSphere Portal to use a remote database
Installing DB2

View information on installing DB2 for use with WebSphere Portal.

Before you begin:

- Make sure DB2 is set up with updated kernel parameters according to the recommendations in the DB2 Quick Beginnings guide at DB2 Technical Support.

- **(For Type 2 drivers only):** DB2 Connect must be installed on the same system as WebSphere Portal, and have the same name as the server profile name.

- When you install DB2 using the DB2 installation program, it automatically creates a DB2 administrative user with the correct operating system rights.

1. If you are using the JDBC driver in type 2 mode, configure your DB2 Connect client with the following commands. If you are using a remote database, complete this step separately from the WebSphere Portal installation.
   - `db2 update dbm cfg using tp_mon_name WAS`
   - `db2 update dbm cfg using spm_name hostname`, where `hostname` is the host name of WebSphere Portal.

   Because the default for `spm_name` is the hostname itself, specifying the `hostname` parameter is optional. If your hostname is more than eight characters, use empty double quotes (" "). For example, `db2 update dbm cfg using spm_name " "`.

2. Before installing DB2, log in with a user ID that has administrative authority.

3. To install DB2, follow the instructions that are provided with the DB2 documentation to install DB2 and the required fix pack.

**Parent topic:** Configuring WebSphere Portal to use DB2

**Next topic:** Create users

**Related Information**

- [DB2 Technical Support](#)
- [DB2 Information Center](#)
Create users

View the steps to set up users for DB2 to work with WebSphere Portal.

Before you begin: You should have completed Installing DB2.

**Prerequisites**

- Installing DB2

For all databases, use one user with administrative rights on the operating system and the DB2 installation. This user can be the database administrative user that is created automatically by the DB2 installation program. This user is the database configuration user that will be used for configuration tasks: creating database tables and performing database transfer. If you choose to have WebSphere Portal also create the databases, the database user should be given SYSADM rights. You only need to manually create an administrator ID when you do not want to use an existing DB2 administrator ID.

A common user name is **db2inst1**, but you can assign any user name as long as it has administrative access and follows the limitations listed here. Do not change the user name after creating it.

The limitations on user names are:

- User names can contain one to eight characters.
- Group and instance names can contain one to eight characters.
- Names cannot be any of the following: **users**
  - **admins**
  - **guests**
  - **public**
  - **local**
- Names cannot begin with: **IBM**
  - **SQL**
  - **SYS**
- Names cannot include accented characters.
- Create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Following are the basic steps to create new users and grant privileges. For more information, refer to the DB2 product documentation.

1. From the DB2 Control Center, expand the object tree until you find the DB Users folder.
2. Right-click the DB Users folder and select **Create/Add**. The Create DB User notebook opens.
3. In the User name field, type a user ID.
4. Assign an authority for the user.
5. Click **OK**.

**Parent topic:** Configuring WebSphere Portal to use DB2

**Previous topic:** Installing DB2

**Next topic:** Creating remote databases
Creating remote databases

A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

Before creating these databases, note the following information:

- Use IBM® DB2 Universal Database™ Enterprise Server Edition JDBC driver, JCC, type 4.
- If the DB2 JDBC type 4 driver is used, only the instructions on the remote server are required. You do not need to install the DB2 client software, and you do not need to complete the steps related to the DB2 client.
- The client software, DB2 Connect, must be correctly configured to connect to the remote DB2 server instance, for example, db2inst1.
- These instructions assume that a remote DB2 server and DB2 Connect are already installed and running. For information on installing DB2 Connect, see the DB2 product documentation.

**Prerequisites**

- Installing DB2
- Create users

To create a database, you must be a DB2 System Administrator with sufficient database privileges (SYSADM or at a minimum SYSCtrl).

1. Log in to the DB2 server system as a user with sufficient database privileges.
2. Initialize a DB2 command environment by opening a command prompt and typing `su - db2inst1`, where `db2inst1` is the database user. The prompt changes to your operating system shell prompt, for example: `$.` In this mode, you must type `db2` at the beginning of each command and use double quotation marks (""") around the entire command. The following example shows how to use the double quotes; it is not an actual command that you run:

   ```
   db2 "CREATE REGULAR TABLESPACE SMS PAGESIZE 4 K MANAGED BY SYSTEM USING ('sms') BUFFERPOOL SMSPOOL"
   ``

   **Note:** If you are using the Command Line Processor (CLP), refer to your DB2 documentation for details. The command prompt is `db2=>`. In this mode, commands can be entered without the `db2` prefix or the double quotation marks. However, the following steps assume you are not using the CLP and are entering commands from your operating system shell prompt, for example: `$.`

3. Run the following commands on the DB2 server system to configure the DB2 database instance:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td>db2set DB2_RA_TO_RS=YES db2set DB2_EVALUNCOMMITTED=YES db2set DB2_INLIST_TO_NLJN=YES db2 &quot;UPDATE DBM CFG USING query_heap_sz 32768&quot; db2 &quot;UPDATE DBM CFG USING maxagents 500&quot; db2 &quot;UPDATE DBM CFG USING sheapthres 0&quot;</td>
</tr>
</tbody>
</table>

4. **(Important)** Failure to complete this step will result in an unsuccessful database transfer. Run the following commands on the DB2 server system to create the necessary databases. **Notes:**

   - Replace `dbname` with the actual name of the database. Run the commands and each time replace `dbname` with the actual values for release, community, customization, Java Content Repository, Feedback, and Likeminds. You will need to run the commands once for each database for a total of six times. **Remember:** DB2 database names cannot...
5. On the DB2 server system, run the following commands. This step is only required for the IBM Java Content Repository database (jcrdb).

- jcrdb is the name of the database used to store user data and objects
- jcr is the jcr user for jcrdb
- dbpassword is the password for the jcr user for the jcrdb

```
db2 "CREATE DB dbname using codeset UTF-8 territory us PAGESIZE 8192"
```

```
db2 "UPDATE DB CFG FOR dbname USING applheapsz 4096"
```

```
db2 "UPDATE DB CFG FOR dbname USING app_ctl_heap_sz 1024"
```

```
db2 "UPDATE DB CFG FOR dbname USING stathheapsz 32768"
```

```
db2 "UPDATE DB CFG FOR dbname USING dbheap 2400"
```

```
db2 "UPDATE DB CFG FOR dbname USING locklist 1000"
```

```
db2 "UPDATE DB CFG FOR dbname USING logfilsiz 4000"
```

```
db2 "UPDATE DB CFG FOR dbname USING logprimary 12"
```

```
db2 "UPDATE DB CFG FOR dbname USING logsecond 20"
```

```
db2 "UPDATE DB CFG FOR dbname USING logbufsz 32"
```

```
db2 "UPDATE DB CFG FOR dbname USING avg_appls 5"
```

```
db2 "UPDATE DB CFG FOR dbname using AUTO_MAINT off"
```

5. Complete the following:

A. On the DB2 server system, run the following commands. This step is only required for the IBM Java Content Repository database (jcrdb).

- jcrdb is the name of the database used to store user data and objects
- jcr is the jcr user for jcrdb
- dbpassword is the password for the jcr user for the jcrdb

```
db2 "CONNECT TO jcrdb USER jcr USING dbpassword"
```

```
db2 "CREATE BUFFERPOOL ICMLSFREQBP4 SIZE 1000 PAGESIZE 4 K"
```

```
db2 "CREATE BUFFERPOOL ICMLSVOLATILEBP4 SIZE 16000 PAGESIZE 4 K"
```

```
db2 "CREATE BUFFERPOOL ICMLSMAINBP32 SIZE 16000 PAGESIZE 32 K"
```

```
db2 "CREATE BUFFERPOOL CMBMAIN4 SIZE 1000 PAGESIZE 4 K"
```

```
db2 "CREATE REGULAR TABLESPACE ICMLFQ32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLFQ32') BUFFERPOOL ICMLSMAINBP32"
```

```
db2 "CREATE REGULAR TABLESPACE ICMLNF32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLNF32') BUFFERPOOL ICMLSMAINBP32"
```

```
db2 "CREATE REGULAR TABLESPACE ICMVFPQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('ICMVFPQ04') BUFFERPOOL ICMLSVOLATILEBP4"
```

```
db2 "CREATE REGULAR TABLESPACE ICMSFQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('ICMSFQ04') BUFFERPOOL ICMLSFREQBP4"
```

```
db2 "CREATE REGULAR TABLESPACE CMBINV04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('CMBINV04') BUFFERPOOL CMBMAIN4"
```

```
ds2 "CREATE SYSTEM TEMPORARY TABLESPACE ICMLSSYSTSPACE32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLSSYSTSPACE32') BUFFERPOOL ICMLSMAINBP32"
```

```
db2 "CREATE SYSTEM TEMPORARY TABLESPACE ICMLSSYSTSPACE4 PAGESIZE 4 K MANAGED BY SYSTEM USING ('ICMLSSYSTSPACE4') BUFFERPOOL ICMLSVOLATILEBP4"
```

```
db2 "DISCONNECT jcrdb"
```

```
db2 "TERMINATE"
```

B. For JDBC Type 2 connections only: On DB2 Connect, use a text editor to open the /etc/services file. If it does not specify the DB2 connection service port, add the following text to specify the port for the remote DB2 instance:

```
db2c_<db2inst1>port1/tcp # DB2 connection service port
```

where <db2inst1> is the name of the DB2 instance ID on the system, and <port1> with the actual port number that is
assigned to the DB2 connection service in your DB2 server installation. The connection service port on the DB2
Client system, WebSphere Portal server, must match the connection service port on the DB2 server. The ports
should match by number but not necessarily by name.

C. Set up the correct service name by entering the following command on the DB2 server system: db2 "UPDATE DBM
CFG USING svcname svce_name" where svce_name is the connection service port name that is specified above.

D. For JDBC Type 2 connections only: On DB2 Connect, set DB2COMM to TCP/IP by using the db2set command
db2set DB2COMM=tcpip.

E. For JDBC Type 2 connections only: Catalog the TCP/IP node with the IP address of the remote database server on
DB2 Connect: db2 "catalog tcpip node remote_db_node_alias remote database_server_node server
collection_service_port" where:
- remote_db_node_alias is the alias name of the database server that you are defining for the WebSphere Application
  Server node name. The alias name can contain one to eight characters.
- database_server_node is the fully qualified host name of your database server system.
- connection_service_port is the name of the DB2 connection service port that is configured in the /etc/services
  file on the database server system.

F. For JDBC Type 2 connections only, catalog the WebSphere Portal databases on DB2 Connect, where:
- remote_db_name_domain, is the cataloged name of the databases on the server system for each domain.
- domain_alias_name, is the database alias names that you are defining.
- remote_db_node_alias is the name that was used previously when you cataloged the TCP/IP node in the previous
  step.

The alias for each database must be different from the actual database name and can only contain up to eight
characters.db2 "catalog db remote_db_name_release as release_alias_name at node remote_db_node_alias"

db2 "catalog db remote_db_name_community as comm_alias_name at node remote_db_node_alias"

db2 "catalog db remote_db_name_customization as cust_alias_name at node remote_db_node_alias"

db2 "catalog db remote_db_name_fdbkdb as fdbkdb_alias_name at node remote_db_node_alias"

db2 "catalog db remote_db_name_lmdb as lmdb_alias_name at node remote_db_node_alias"

db2 "catalog db remote_db_name_jcrdb as jcrdb_alias_name at node remote_db_node_alias"

G. For JDBC Type 2 connections only: Log out of DB2 Connect by entering db2 "terminate".

H. For JDBC Type 2 connections only, on DB2 Connect, test your remote connection by issuing the following command
in the DB2 command window: db2 "connect to alias_name user username using password", where alias_name
is the alias name that you defined above, username is the database user, and password is the password assigned to
the database user.

Parent topic: Configuring WebSphere Portal to use DB2
Previous topic: Create users
Next topic: Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the \wp_profile_root\PortalServer/config/tablespaces/ddomain.space_mapping.properties file.
- The page size of table spaces used by WebSphere Portal must be 8192 bytes.
- For details on creating table spaces refer to the documentation for the database.

**Prerequisites**
- Installing DB2
- Create users
- Creating remote databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file \wp_profile_root\PortalServer/config/tablespaces/ddomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - ddomain.table_name.tablespace
   - ddomain.table_name.index_name.indexspace
   For the file name and each table space and index space property pair, ddomain can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword IN and a space. For example: community.COMP_INST.tablespace=IN COMM8KSPACE
   Repeat this step for each domain that you are transferring.
4. Save and close ddomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
   - UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

**Parent topic:** Configuring WebSphere Portal to use DB2  
**Previous topic:** Creating remote databases  
**Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

Tip: For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Working with properties files:
- The WebSphere Portal database can be used to hold information for applications such as Feedback and Like Minds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:

  - release.DbName=release
  - jcr.DbName=jcrdb
  - feedback.DbName=fdbkdb
  - likeminds.DbName=lmdb
  - community.DbName=commdb
  - customization.DbName=custdb

  - If you are using a remote database, enter the values for the remote server.
  - Use a forward slash (/) instead of a backslash (\).
  - There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
  - The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
  - Depending on which database domain has to be configured, replace dbdomain with:
    - release
    - customization
    - community
    - jcr
    - feedback
    - likeminds

  - The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
- `dbname DbType`
- `dbname DbName`
- `dbname DbUrl`
- `dbname DbSchema`

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If `DbUser`, `DbUrl`, and `DbPassword` are not the same across domains, the value for `DataSourceName` must differ from the `DataSourceName` of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   - `wprofile_root/ConfigEngine/properties/wkplc.properties`
   - `wprofile_root/ConfigEngine/properties/wkplc_comp.properties`
   - `wprofile_root/ConfigEngine/properties/wkplc_dbtype.properties`

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In `wkplc_comp.properties`, most properties are repeated for each domain.

2. Use a text editor to open the properties file `wkplc_comp.properties` and modify the values to correspond to your environment.
   A. For `dbname DbType`, type `db2`.
   B. For `dbname DbName`, type the name of the WebSphere Portal domain database. **Notes:**
      - This value is also the database element in the `dbname DbUrl` property.
      - This value is the TCP-IP alias for the database.
   C. For `dbname DbSchema`, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   D. For `dbname DataSourceName`, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For `dbname DbUrl`, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:** The database element of this value should match the value of `DbName`.
   F. For `dbname DbUser`, type the user ID for the database administrator.
   G. For `dbname DbPassword`, type the password for the database administrator.
   H. Optional: For `dbname DbRuntimeUser`, type the user ID of the database user that should be used by WebSphere Portal to connect to the database at runtime.
   I. If `dbname DbRuntimeUser` is specified, you must set `dbname DbRuntimePassword` to be the password of the runtime database user.
   J. For `dbname DBA.DbUser`, type the database administrator user ID for privileged access operations during creation of the database.
K. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.
L. For `dbdomain.XDbName`, type the database loop back alias that needs to be set if you plan to use the `create-database` task. **Note:** Required only for local databases using a JDBC Type 2 driver.
M. For `dbdomain.DbNode`, type the value for the node database. Set this value if you want to call `create-database`. **Note:** Required only for local databases.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For `source.domain DbType`, type of the database you are currently configured to use. The value for `source.domain DbType` is Derby by default.
B. For `source.domain DbName`, type the name of the database domain you are currently using.
C. For `source.domain DbSchema`, type current schema identifier for objects within the database for this domain.
D. For `source.domain DataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.
E. For `source.domain DbUrl`, type the url currently used to access your database.
F. For `source.domain DbUser`, type the name of the user accessing this database.
G. For `source.domain DbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

A. For `db2.DbDriver`, type the name of the JDBC driver class.
B. For `db2.DbLibrary`, type the directory and name of the `.zip` or `.jar` file that contains the JDBC driver class.
C. For `db2.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Configuring WebSphere Portal to use DB2  
**Previous topic:** Assigning custom table spaces  
**Next topic:** Configuring JCR collation support
Configuring JCR collation support

View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties

1. Stop the WebSphere Portal server:
   ```
   ./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password
   ```

2. If using a remote DB2 server, copy the following files from the WebSphere Portal server to a temporary directory on the remote DB2 server:
   ```
   PortalServer_root/jcr/prereq.jcr/config/collation.jar
   PortalServer_root/jcr/prereq.jcr/config/registerCollationUDFTemplate.sql
   ```

3. Set up collation on the database where the JCR domain is located.
   - Change to the directory `db2_instance_owner_home`/sqlib/function.
     ```
     Remote DB2:
     db2_instance_owner_home/sqlib/java/jdk/bin/jar -xvf temporary location/collation.jar icm/CollationUDF.class
     ```
     ```
     Local DB2:
     db2_instance_owner_home/sqlib/java/jdk/bin/jar -xvf PortalServer_root /jcr/prereq.jcr/config/collation.jar icm/CollationUDF.class
     ```
   - If using a local DB2, change to the directory `wp_profile_root`PortalServer/jcr/config. If using a remote DB2, change to the temporary directory where you copied the files in a previous step.

4. Open the file `registerCollationUDFTemplate.sql` and change all `SCHEMA` references to the JCR schema; for example, `JCR`. The value set for `SCHEMA` should match the value set for the `jcr.DbSchema` property. You specify `jcr.DbSchema` in the configuration file `wkplc_comp.properties` when you modify database properties.

5. Connect to the JCR database by running `db2 connect to <jcrdb> user <userid> using <password>`. You must connect to the JCR database using the userid and password of the WebSphere Portal JCR data source.

6. Execute the command `values schema.sortkeyj('abc','en')` in the previous step. If the command completes successfully, the UDF is registered correctly as follows:

7. Edit the `icm.properties` file, located in `wp_profile_root`PortalServer/jcr/lib/com/ibm/icm directory. Add the following section to the end of the file:

```
# Enable/Disable collation support for all DB2 platforms
```

---

**Operating systems:** AIX, HP-UX, i5/OS, Linux, Solaris, Windows
# Disabled by default
jcr.query.collation.db2.enabled = true

# Database specific collation mappings
# These mappings apply a Java locale name into a collation name
# supported by the underlying database.
# Example mappings for DB2 platform

# English
jcr.query.collation.en = en

# Swedish
jcr.query.collation.sv = sv

jcr.query.collation.zh = zh
jcr.query.collation.de = de
jcr.query.collation.da = da
jcr.query.collation.hu = hu
jcr.query.collation.jp = jp

6. Enter the following command to start the WebSphere Portal server:

   . /startServer.sh WebSphere_Portal

**Parent topic:** Configuring WebSphere Portal to use DB2  
**Previous topic:** Modifying database properties  
**Next topic:** Configure WebSphere Portal to use DB2
Configure WebSphere Portal to use DB2

View the steps to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties

Tips:

- To run these tasks as a non-root user, you must first run the task chown -R non-root_user WebSphereDir.
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

Steps for transferring data to another supported database

1. If you are running a type 2 connection, before transferring data edit the db2cli.ini file. Failure to follow these steps will cause the database transfer to hang at the task action-process-constraints.
   A. Locate the file /home/db2inst1/sqllib/cfg/db2cli.ini.
   B. Add the following to the end of the file. Leave an empty line after ReturnAliases=0.[COMMON]
      
      [COMMON]
      DYNAMIC=1
      ReturnAliases=0

2. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.

3. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2. .

   /home/db2inst1/sqllib/db2profile
   where db2inst1 represents your database instance

   Note: You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.

4. Enter the following commands to validate configuration properties...

   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password

   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password

5. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.
6. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td><code>./stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td><code>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

7. Transfer the database:

   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory `wp_profile_root/ConfigEngine`.

   B. Enter the following command: `./ConfigEngine.sh database-transfer -DWasPassword=password`

   **Note:**
   - To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command: `./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password`
   - If you have been storing data in IBM® Derby for a long time, database transfer could fail with OutOfMemory exceptions. If database transfer fails, add the following property to the command in this step: `./ConfigEngine.sh database-transfer -DDbtJavaMaxMemory=1536M -DWasPassword=password`

   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.
   If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

8. After transferring the database tables, perform a reorg check to improve performance. Perform this step for each database alias in the property file.

A. Connect to a database with the following command: `db2 connect to database_alias user db2admin_userid using password`

   **Note:** Additional options might be required if additional security has been installed. Refer to DB2 Universal Database commands by example for links to the command reference.

B. After it is connected, run the following commands from the DB2 prompt: `db2 reorgchk update statistics on table all > xyz.out`

C. Look in the reorg column for entries marked with a * (star or asterisk) in the file `xyz.out`. For each line with a *, note the `tablename` and run the following command for each `tablename`: `db2 reorg table tablename db2 terminate db2rbind database_name -l db2rbind.out -u db2_admin -p password`

D. The output file `db2rbind.out` is only created when there is an error for the `db2rbind` command.

9. Change to the directory `wp_profile_root/bin`.

10. Enter the following command to start the WebSphere Portal server: `./startServer.sh WebSphere_Portal`
Configuring DB2 for large file handling in Web Content Management

If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

**Prerequisites**
- Installing DB2
- Create users
- Creating remote databases
- Modifying database properties
- Configure WebSphere Portal to use DB2

**Note:** You only need to perform these steps if you are using Web Content Management.

1. Log into the WebSphere Application Server administrative console.
2. Click **Resources > JDBC > Data sources**.
3. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
4. Select the name of the data source that is defined in `wkplc_comp.properties` for the JCR database domain. The default data source is `wpdbDS`.
5. Click **Custom properties**.
6. Ensure that the `fullyMaterializeLobData` property is set to `false`.

**Parent topic:** Configuring WebSphere Portal to use DB2

**Previous topic:** Configure WebSphere Portal to use DB2

**Next topic:** Changing driver types
Changing driver types

WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Before you begin, ensure that the following conditions are met:

- The WebSphere Portal database has been successfully transferred to DB2 using the `database-transfer` configuration task.
- The files `wkplc_comp.properties` and `wkplc_dbtype.properties` have been modified to set the correct values for the DB2 drivers that you are switching to:

  In the file `wkplc_comp.properties` set each `<Domain>.DbUrl` property using the following formats:
  
  ```
  # db2 (type 2):
  jdbc:db2:wpsdb
  
  # db2 (type 4):
  jdbc:db2://<YourDatabaseServer>:50000/wpsdb:returnAlias=0;
  ```

  In the file `wkplc_dbtype.properties` set the `db2.DbLibrary` property using the following format:

  ```
  # For DB2 Type 2
  <SQLLIB>/java/db2java.zip
  
  # For DB2 Type 4 driver use
  <SQLLIB>/java/db2jcc.jar:<SQLLIB>/java/db2jcc_license_cu.jar
  ```

  In the file `wkplc_dbtype.properties` set the `db2.DbDriver` property using the following format:

  ```
  # For DB2 Type 2
  COM.ibm.db2.jdbc.app.DB2Driver
  
  # For DB2 Type 4 driver use
  com.ibm.db2.jcc.DB2Driver
  ```

**Note:** If WebSphere Portal is installed on the same machine as the DB2 server and you switch from a JDBC Type 4 connection to a JDBC Type 2 connection, verify that you have created the alias names for the DB2 databases as described in `Creating remote databases` and that the alias names are specified for the databases in the file `wkplc_comp.properties`. When switching from a JDBC Type 2 connection to a JDBC Type 4 connection, remove the database alias names and refer to the databases directly. This is required because of a limitation in the DB2 Universal JDBC driver.

**Prerequisites**

- Installing DB2
- Create users
- Creating remote databases
- Modifying database properties
- Configure WebSphere Portal to use DB2
- Configuring DB2 for large file handling in Web Content Management

1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.
2. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2.

   ```
   /home/db2inst1/sqllib/db2profile
   ```

   where `db2inst1` represents your database instance.

   **Note:** You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.
3. Enter the following commands to validate configuration properties.

   ```
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ```
4. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td><code>/stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td><code>/stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

6. Change to the directory `wp_profile_root/ConfigEngine`.

7. To change from one supported driver to the other, run the following task to connect the database, including only the domains that require the switch:

```
```

8. Change to the directory `wp_profile_root/bin`.

9. Enter the following command to start the WebSphere Portal server:

```
./startServer.sh WebSphere_Portal
```

Parent topic: Configuring WebSphere Portal to use DB2
Previous topic: Configuring DB2 for large file handling in Web Content Management
Configuring WebSphere Portal to use DB2 for z/OS

View information on installing and setting up DB2 for z/OS to work with WebSphere Portal.

1. Installing DB2 for z/OS
   View information on how to install DB2 for z/OS for use with IBM® WebSphere® Portal.

2. Creating users
   Complete this procedure to set up users for IBM DB2 Universal Database™ for z/OS® to work with IBM WebSphere Portal.

3. Creating remote databases
   A remote database resides on a different machine than WebSphere Portal. You must manually create the required databases before configuring WebSphere Portal to work with DB2 for z/OS.

4. Optional: Creating the Java Content Repository database
   View information on setting up your Java Content Repository database to work with WebSphere Portal.

5. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

6. Configuring WebSphere Portal to use DB2 for z/OS
   View information on manually transferring data to the DB2 for z/OS you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository to DB2 for z/OS. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configure WebSphere Portal to use a remote database
Installing DB2 for z/OS

View information on how to install DB2 for z/OS for use with IBM® WebSphere® Portal.

The following prerequisites must exist separately from the WebSphere Portal installation:

- If you are using the DB2 Legacy Type 2 JDBC driver, configure your DB2 Connect client with the following commands:
  
  - `db2 update dbm cfg using tp_mon_name WAS`
  - `db2 update dbm cfg using spm_name hostname`, where `hostname` is the host name for the machine where the DB2 Connect client is installed (same as portal machine).

- If you are using the DB2 Legacy Type 2 JDBC driver, enable extended shared memory usage with the following commands:
  
  - `export EXTSHM=ON`
  - `db2set DB2ENVLIST=EXTSHM`
  - `db2start`

  To make the change permanent, you must add the environment variable by adding the following to the `profile.env` file:

  ```
  DB2ENVLIST='EXTSHM'
  ```

  in `/home/db2inst/sqllib/userprofile`

  add: `export EXTSHM=ON`

  **Note:** The shell must be reopened before restarting DB2 for z/OS.

- Both type 2 and type 4 drivers are supported. If you are using the DB2 Legacy Type 2 JDBC driver, the DB2 Connect client must be installed on the same machine as WebSphere Portal to connect to the remote database.

- The DB2 subsystem must be on a supported z/OS platform.

- Update the BP8K0 catalog bufferpool to 35,000 buffers before performing database transfer. You should change this value based on your environment. The SYSIBM.SYSDATABASE table resides in this bufferpool and is used extensively by DB2 for z/OS during database transfer.

To use DB2 for z/OS as the database software for WebSphere Portal, you must have DB2 for z/OS installed on your z/OS system. Refer to the DB2 for z/OS product documentation for instructions. Refer to the Planning for DB2 for z/OS topic for additional considerations for configuring DB2 for z/OS as the WebSphere Portal database.

**Parent topic:** Configuring WebSphere Portal to use DB2 for z/OS  
**Next topic:** Creating users
Creating users

Complete this procedure to set up users for IBM® DB2 Universal Database™ for z/OS® to work with IBM WebSphere® Portal.

Before you begin: You should have completed Installing DB2 for z/OS.

- Prerequisites

- Installing DB2 for z/OS

- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

- If you are configuring multiple WebSphere Portal instances to use a single DB2 for z/OS subsystem, you must create a unique database user for each WebSphere Portal instance. By default all database table names include the name of the database user used to access the data. Therefore, to prevent table name conflicts, you must create a unique database user for each WebSphere Portal instance on the shared DB2 for z/OS subsystem.

- Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Use the following steps to grant access permissions to the database users. Repeat these steps for each WebSphere Portal instance you are setting up.

1. Create all database user IDs in the security product you are using on z/OS. For jcrschema, the database schema name for IBM Java Content Repository data, create a group and connect it to the database user ID for Java Content Repository data, jcr. The following sample shows the RACF definition of such a user ID and group, where jcr is the database user ID for Java Content Repository data, yourDefaultUserGroup is your default RACF group for database user IDs, jcrschema is the database schema name for Java Content Repository data, and yourDefaultGroup is your default RACF group for groups. If you have some other security product such as Top Secret or ACF2 instead of RACF, translate the sample RACF definition into the appropriate syntax before running the job.

   ```
   ADDUSER jcr DFLTGRP(yourDefaultUserGroup) NAME('WAS DB2 ACCESS USER')
   PW USER(jcr) NOINTERVAL
   ALU jcr PASSWORD(********) NOEXPIRED
   ADDGROUP jcrschema SUPGROUP(yourDefaultGroup)
   CONNECT jcr GROUP(jcrschema)
   ```

2. Run the following SQL statements using a tool like SPUFI while logged on to the DB2 subsystem to grant appropriate rights on the newly created databases. If you are configuring multiple WebSphere Portal instances to use a single DB2 for z/OS subsystem, be sure to use the database user associated with the WebSphere Portal instance you are setting up. (C) create/alter tablespaces

   (C) create/alter tables

   (C) create/alter indice;

   (C+R) read/write data

   (C) - at configuration time

   (R) - at runtime
GRANT USE OF ALL BUFFERPOOLS TO releaseusr;
GRANT USE OF ALL BUFFERPOOLS TO communityusr;
GRANT USE OF ALL BUFFERPOOLS TO customizationusr;

GRANT DBADM ON DATABASE jcrdbnameonzos TO jcr;
GRANT USE OF ALL BUFFERPOOLS TO jcr;
GRANT DBADM ON DATABASE fdbkdbnameonzos TO feedback;
GRANT USE OF ALL BUFFERPOOLS TO feedback;
GRANT DBADM ON DATABASE lmdbnameonzos TO lmdbus;
GRANT USE OF ALL BUFFERPOOLS TO lmdbus;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO releaseusr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO communityusr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO customizationusr;

GRANT SELECT ON SYSIBM.SYSTABLES TO releaseusr;
GRANT SELECT ON SYSIBM.SYSTABLES TO communityusr;
GRANT SELECT ON SYSIBM.SYSTABLES TO customizationusr;

GRANT SELECT ON SYSIBM.SYSCOLUMNS TO jcr;
GRANT SELECT ON SYSIBM.SYSTABLES TO jcr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO fdbkdbus;
GRANT SELECT ON SYSIBM.SYSTABLES TO fdbkdbus;
GRANT SELECT ON SYSIBM.SYSTABLES TO lmdbus;
GRANT SELECT ON SYSIBM.SYSTABLES TO lmdbus;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO releaseusr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO communityusr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO customizationusr;

GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO jcr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO fdbkdbus;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO lmdbus;
GRANT SELECT ON SYSIBM.SYSRELS TO releaseusr;
GRANT SELECT ON SYSIBM.SYSRELS TO communityusr;
GRANT SELECT ON SYSIBM.SYSRELS TO customizationusr;

GRANT SELECT ON SYSIBM.SYSRELS TO jcr;
GRANT SELECT ON SYSIBM.SYSRELS TO fdbkdbus;
GRANT SELECT ON SYSIBM.SYSRELS TO lmdbus;
GRANT USE OF STOGROUP jcrstogroup TO jcr;
GRANT CREATEIN, DROPIN ON SCHEMA jcrschema TO jcr;
GRANT SELECT ON SYSIBM.SYSTABLESPACE TO jcr;
GRANT SELECT ON SYSIBM.SYSVIEWS TO jcr;
GRANT SELECT ON SYSIBM.SYSDUMMY1 TO jcr;
GRANT SELECT ON SYSIBM.SYSTRIGGERS TO jcr;
GRANT SELECT ON SYSIBM.SYSINDEXPART TO jcr;
GRANT SELECT ON SYSIBM.SYSINDEXES TO jcr;
GRANT SELECT ON SYSIBM.SYSSYNONYMS TO jcr;

Where:
* releasenameonzos, communitynameonzos, customizationnameonzos, and releaseusr, communityusr, customizationusr represent the databases and database users, respectively, of the WebSphere Portal instance you are setting up. (These users must be created on the host system.)

* jcrdbnameonzos and jcr are the database and database user, respectively, for Content Repository data.

* fdbkdbnameonzos and feedback are the database and database user, respectively, for Feedback data.

* lmdbnameonzos and lmdbusr are the database and database user, respectively, for Likeminds data.

* jcrschema is the database schema name for Content Repository data.

3. Grant the necessary access rights to all users who might require them. Depending on the architecture that you choose, these users might include Java Content Repository and Feedback users.

**Parent topic:** Configuring WebSphere Portal to use DB2 for z/OS

**Previous topic:** Installing DB2 for z/OS

**Next topic:** Creating remote databases
Creating remote databases

A remote database resides on a different machine than WebSphere Portal. You must manually create the required databases before configuring WebSphere Portal to work with DB2 for z/OS.

Before you begin, complete the following prerequisites:

- These instructions assume that you will use a local IBM® DB2 Universal Database™ Enterprise Server Edition Connect to connect to a remote IBM DB2 Universal Database for z/OS® server, and that IBM WebSphere® Application Server, WebSphere Portal, and DB2 Connect will be installed on the same machine.
- To run SQL statements, you can use a tool like SPUFI. The specified statements include CREATE STOGROUP, CREATE DATABASE and CREATE TABLESPACE. For CREATE STOGROUP you have to replace the icmvolumes and icmvcat variables with volume serial numbers and a catalog name for your environment. For CREATE TABLESPACE you can specify a specific BUFFERPOOL. Refer to the DB2 for z/OS SQL Reference for a description of these options.
- Ensure that a TEMP database is created for your subsystem. You can use the following statements to create a TEMP database:

```
CREATE DATABASE db_name AS TEMP;
CREATE TABLESPACE ts_name IN db_name;
```

- Log on to the DB2 subsystem on the host server. DB2 system administrator rights are needed to create the databases.
- Replace variables as follows:
  - releasenameonzos, communitynameonzos, and customizationnameonzos are the WebSphere Portal databases for WebSphere Portal and Member Manager data.
  - fdbkdbnameonzos and fdbkdtbs are the database and table space, respectively, for Feedback data.
  - lmdbnameonzos and lmdtbs are the database and table space, respectively, for LikeMinds data.
- Because large objects are stored in columns that can become very large, logging changes to these columns requires a huge amount of log space. For this reason, large object (LOB) logging is disabled by default for tablespaces that contain such data. With LOB logging disabled, you can recover full backups, but not incremental backups that can be used for point in time recovery. To recover point in time backups, you must enable LOB logging. For detailed instructions, see Technote 1306637, Managing LOB logging in DB2 for z/OS.

- Prerequisites
  - Installing DB2 for z/OS
  - Creating users

Use the following steps to set up WebSphere Portal with a remote instance of DB2 for z/OS. Refer to Planning for DB2 for z/OS for a list of databases and table space names. If you are configuring multiple WebSphere Portal instances to use a single DB2 subsystem, be sure to use the database and table space names associated with the WebSphere Portal instance you are setting up.

1. CREATE DATABASE releasenameonzos CCSID UNICODE;
2. CREATE DATABASE communitynameonzos CCSID UNICODE;
3. CREATE DATABASE customizationnameonzos CCSID UNICODE;
4. Execute the steps in the topic Creating the Java Content Repository database.
5. CREATE DATABASE fdbkdbnameonzos CCSID UNICODE;
6. CREATE TABLESPACE fdbkdtbs IN fdbkdbnameonzos USING STOGROUP SYSDEFLT PRIQTY 5000 SECQTY 500 LOCKSIZE ROW;
7. CREATE DATABASE lmdbnameonzos CCSID UNICODE;
8. CREATE TABLESPACE lmdbts IN lmdbnameonzos USING STOGROUP SYSDEFLT PRIQTY 5000 SECQTY 500;

**Parent topic:** Configuring WebSphere Portal to use DB2 for z/OS  
**Previous topic:** Creating users  
**Next topic:** Creating the Java Content Repository database

**Related Information**

- Managing LOB logging in DB2 for z/OS
Creating the Java Content Repository database

View information on setting up your Java Content Repository database to work with WebSphere Portal. First, you need to create multiple databases for scalability.

The IBM Java Content Repository database grows in size as the IBM® WebSphere® Portal server is used. To support this growth, which includes the dynamic creation of many new tables within the database, the WebSphere Portal server allows you to create multiple databases within a single IBM DB2 Universal Database™ for z/OS® subsystem. A minimum of two databases is recommended, but more can be created if necessary. A single database setup is not recommended; it will quickly become constrained. See the Performance guides on the product documentation Web site for more information.

Each database that is created must begin with a common prefix; there is no convention required for the remainder of the name. For example, to create xx number of databases, you may choose to use the following commands:

```
CREATE DATABASE JCRDB01
CREATE DATABASE JCRDB02
...
CREATE DATABASE JCRDBxx
```

In this case, JCR is the common prefix. You also have to select a maximum number of user-defined tables (UDT) that each database will be allowed to hold; the recommended value is 400.

- **Prerequisites**
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases

A sample script of the commands that you can use to create the Java Content Repository database in your DB2 for z/OS installation is shown below. The sample demonstrates the creation of a single database. To follow the recommendation of creating at least two databases, duplicate the commands for each additional database as indicated below. Find a template of these commands in the file `PortalServer_root` installer/wp.config/config/templates/db/db2_zos/jcr_sample.sql.

**Notes:**
- It is not necessary to duplicate every tablespace definition in every database.
- In order to run the commands shown in the sample, you must know the database name and the IDs of the MVS volumes where the Java Content Repository database tables will be stored.
- Edit the template file for your installation environment before running the commands shown in the sample:
  - Replace `jcrdbnameX` with the name of the database. Remember that each database name must begin with a common prefix.
  - Replace `stogroup` with the name of your storage group.
  - Replace `icmvolumes` with the IDs of the MVS volumes where the Java Content Repository database tables will be stored. These values are assigned by the database administrator.
  - Replace `icmvcat` with the name of the virtual catalog.
  - Replace `jcr` with the name of database user ID.
  - Replace `4kbp` with the name of your 4K bufferpool.
  - Replace `32kbp` with the name of your 32K bufferpool.
  - Replace `jcrschema` with the schema name of your Java Content Repository domain.
--DROP DATABASE jcrdbnameX
--DROP STOGROUP stogroup
--DROP ALIAS jcrschema.ICMFICTITIOUS

CREATE STOGROUP stogroup VOLUMES (icmvolumes) VCAT icmvcat

GRANT USE OF STOGROUP stogroup to jcr

CREATE DATABASE jcrdbnameX STOGROUP stogroup BUFFERPOOL 4kbp INDEXBP 4kbp CCSID UNICODE

Note: The following two tablespace commands must be created in every database:

CREATE TABLESPACE ICMLFQ32 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICMTSQ04 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

Note: The following tablespaces are required in the first database only:

CREATE TABLESPACE ICMVFQ04 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICMSFQ04 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSTADO IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRIDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRSCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRISLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRGCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRIGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSTDPV IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSDOAC IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RESMLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COLNLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USERLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USEGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE SYSCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
1343
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CACLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CPERLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NLSLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE MAXKLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NLSKLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITTDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITVDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITVILSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COMDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COMALSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COMFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COVDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COVALSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITALLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE IT11LSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE IV11LSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE LI11LSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITTRLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RMATLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CHEOLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE MIMTLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITEELSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
1344


CREATE TABLESPACE TSINJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE TSERJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE TIMEJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE SPRTJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE RWSJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE NODEJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ROOTJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE PRPDJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE NSURJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE NSPRJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE NODRJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE NODTJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE NODDJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE MAXSJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE

CREATE TABLESPACE LINKJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE GLBPJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE GENDJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE DWSLJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ADDPJRTS IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ICUT601 IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ICUT341 IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ICUT331 IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ICUT321 IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ICUT311 IN
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

CREATE TABLESPACE jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TSPEJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TSTAJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RIDSJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE WSN0JRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE LOB TABLESPACE ICMLOBT1 IN jcrdbnameX LOG NO
CREATE LOB TABLESPACE ICMLOBT2 IN jcrdbnameX LOG NO
CREATE LOB TABLESPACE ICMLOBT3 IN jcrdbnameX LOG NO
CREATE ALIAS jcrschema.ICMFICTITIOUS FOR SYSIBM.SYSDUMMY1
CREATE TABLESPACE DLKRJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE LKRLJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

Parent topic: Configuring WebSphere Portal to use DB2 for z/OS
Previous topic: Creating remote databases
Next topic: Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.
- When using IBM® DB2 Universal Database™ for z/OS® as a data store, WebSphere Portal requires that its indexes are not padded. Therefore, you must set the DSNZPARM parameters to RETVLCFK=NO or PADIX=NO, or both.

**Prerequisites**
- Installing DB2 for z/OS
- Creating users
- Creating remote databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.index_name.indexspace
   For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds
3. Assign a table space to each .tablespace entry in the mapping file. Assignments to .indexspace entries are ignored. The table space name must be qualified by the database name and prepended by the keyword IN and a space. For example: community.COMP_INST.tablespace=IN COMM8KSPACE
   Repeat this step for each domain that you are transferring.
4. Save and close dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
- UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

**Parent topic:** Configuring WebSphere Portal to use DB2 for z/OS

**Previous topic:** Creating the Java Content Repository database

**Next topic:** Configuring WebSphere Portal to use DB2 for z/OS
Configuring WebSphere Portal to use DB2 for z/OS

View information on manually transferring data to the DB2 for z/OS you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository to DB2 for z/OS. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.

- Prerequisites
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases

Steps for transferring data to another supported database

1. If you are running a type 2 connection, before transferring data edit the db2cli.ini file. Failure to follow these steps will cause the database transfer to hang at the task action-process-constraints.

   A. Locate the file /home/db2inst1/sqllib/cfg/db2cli.ini.
   B. Add the following to the end of the file. Leave an empty line after returnAliases=0.

```
[COMMON]
DYNAMIC=1
ReturnAliases=0
```

2. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.

3. Enter the following commands to validate configuration properties...

```
./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
```

4. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td>./stopServer.sh server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

6. Transfer the database:

   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory wp_profile_root/ConfigEngine.

   B. Enter the following command:

```
./ConfigEngine.sh database-transfer -DWasPassword=password
```

**Note:** To select specific database domains to transfer, modify the -DTransferDomainList specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter...
the following command: 

```
./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
```

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

If the configuration fails, verify the values in the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files and then repeat this step.

7. From the command line processor, reset the check pending status on the WebSphere Portal table spaces listed here.

CHECK DATA is a DB2 batch utility that is invoked through JCL or through the DB2 utility panels. Type the following utility commands to check pending status:

- `check data tablespace releasenameonzos.TS320A`
- `check data tablespace releasenameonzos.TS280A`
- `check data tablespace releasenameonzos.TS300A`
- `check data tablespace releasenameonzos.TS2110A`
- `check data tablespace releasenameonzos.TS830A`
- `check data tablespace communitynameonzos.TS8000B`
- `check data tablespace communitynameonzos.TS8011B`
- `check data tablespace communitynameonzos.TS280B`
- `check data tablespace customizationnameonzos.TS2110C`
- `check data tablespace jcrdbnameonzos.ICMSPQ04`
- `check data tablespace jcrdbnameonzos.TS2110D`

where releasenameonzos, communitynameonzos, and customizationnameonzos are the names of your WebSphere Portal databases, and jcrdbnameonzos is the name of your JCR database. Refer to the Utility Guide and Reference for DB2 for z/OS for additional details.

8. Run the RUNSTATS utility as shown in the following example:

```
LISTDEF JCRDBZOS INCLUDE TABLESPACE JCRDBZOS.* BASE
RUNSTATS TABLESPACE LIST JCRDBZOS INDEX(ALL) KEYCARD TABLE(ALL)
LISTDEF JCRDB01 INCLUDE TABLESPACE JCRDB01.* BASE
RUNSTATS TABLESPACE LIST JCRDB01 INDEX(ALL) KEYCARD TABLE(ALL)
...
```

9. Start the WebSphere Application Server.

10. Verify that the WebSphere Portal application server is running by opening the following URL in a browser:

```
http://hostname.companyname.com:port_number/wps/portal
```

where hostname.companyname.com is the fully qualified host name of the machine where WebSphere Portal is running and port_number is the transport port that is created by WebSphere Application Server.

If you have performed database transfer in a clustered environment, you must explicitly copy the `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm/icm.properties` file from the primary node on which the database-transfer task was processed to all secondary nodes. This ensures that the secondary nodes have the new JCR database values specified in the icm.properties file.

1. Stop the portal server on the secondary node.
2. Copy the `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm/icm.properties` file from the primary node and replace the icm.properties file on the secondary node with the new file from the primary node.
3. Start the portal server on the secondary node.
Configuring WebSphere Portal to use Oracle

View information on installing and setting up Oracle to work with WebSphere Portal.

1. Installing Oracle
   View information on how to install Oracle for use with WebSphere Portal.

2. Creating users
   View the steps to set up users for Oracle to work with WebSphere Portal.

3. Creating databases
   View some important considerations before setting up Oracle databases to work with WebSphere Portal.

4. Modifying database properties
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

6. Creating JCR table spaces
   This topic provides manual instructions for creating JCR table spaces for Oracle.

7. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

8. Configure WebSphere Portal to use Oracle
   This section provides information on how to manually transfer data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configure WebSphere Portal to use a remote database
Installing Oracle

View information on how to install Oracle for use with WebSphere Portal.
You can use Oracle as the database software. You must install Oracle separately and before performing a database transfer.
Refer to Oracle product documentation for installation instructions.

**Parent topic:** Configuring WebSphere Portal to use Oracle

**Next topic:** Creating users
Creating users

View the steps to set up users for Oracle to work with WebSphere Portal.

Before you begin:
- You should have installed Oracle.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

**Prerequisites**
- Installing Oracle

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Start the Oracle SQL command line processor 'SQL*Plus' by entering `sqlplus /nolog` on the operating system command prompt
2. Log in to the Oracle database by executing command `connect user/password@database name`. For example: `system/manager@wpsdb` will log the administrative user system with a password of manager into the wpsdb database.
3. Create the WebSphere Portal user `dbdomain.DbUser`, where `dbdomain` is replaced by release, community, or customization.
   A. Log in to the database in which you want to create the new users.
   B. Create the users that you need in the appropriate databases by entering the following commands in the SQL*Plus tool, where `user_tablespace` is the default tablespace that is identified by the database administrator to store user objects and `temp_tablespace` is identified to store temporary objects. All names must be in lower case.

   ```sql
   SQL> create user releaseusr identified by password
       default tablespace user_tablespace
       temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO releaseusr;
   SQL> create user communityusr identified by password
       default tablespace user_tablespace temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO communityusr;
   SQL> create user customizationusr identified by password
       default tablespace user_tablespace temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO customizationusr;
   SQL> create user jcr identified by password
       default tablespace user_tablespace temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO jcr;
   ```
4. Connect to the Feedback database:
   A. Enter the following command: `SQL> connect`
B. Enter user-name: username/password@dbname where 'username' is an existing administrative user in the database. For example, system/manager@fdbkdb will log the administrative user system with a password of manager into the fdbkdb database.

C. Create the Feedback user:

```
SQL> create user feedback identified by password
    
default tablespace user_tablespace
    
temporary tablespace temp_tablespace;
    
GRANT UNLIMITED TABLESPACE TO feedback
```

D. Log out of the command line tool using the command SQL> exit.

5. Connect to the LikeMinds database:

A. Enter the following command:
```
SQL> connect
```

B. Enter user-name: username/password@dbname, where 'username' is an existing administrative user in the database. For example, system/manager@lmdb will log the administrative user system with a password of manager into the lmdb database.

C. Create the LikeMinds user:

```
SQL> create user lmdbusr identified by password
    
default tablespace user_tablespace
    
temporary tablespace temp_tablespace;
    
GRANT UNLIMITED TABLESPACE TO lmdbusr
```

D. Log out of the command line tool using the command SQL> exit.

6. Perform the following steps to verify that you can connect to the database with the user ID and password that you just created:

A. Enter the SQL> connect command to connect to the content database.

B. Enter user-name: username/password@dbname, where 'username' is an existing administrative user in the database. For example, system/manager@jcrdb will log the administrative user system with a password of manager into the Oracle database.

**Parent topic:** Configuring WebSphere Portal to use Oracle

**Previous topic:** Installing Oracle

**Next topic:** Creating databases
Creating databases

View some important considerations before setting up Oracle databases to work with WebSphere Portal.

For information about creating databases, refer to the Oracle product documentation. For information on the recommended database architecture and the databases you will need to create, see the Planning for Oracle topic. Be sure that all databases to be used with WebSphere Portal are created as UNICODE character set databases.

If you are using remote Oracle 9 or 10 databases, you must also copy the ojdbc14.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM® WebSphere® Application Server Version 7.0, you must also copy the ojdbc6.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM WebSphere Application Server Version 6.1, you must also copy the ojdbc5.jar file from the remote Oracle server to the WebSphere Portal machine. The typical location is the oracle_home/sqldeveloper/jdbc/lib directory. Record the copy location on your local machine for future reference.

When creating Oracle databases for use with WebSphere Portal, you should consider the following information:

- All databases must be created manually before configuring WebSphere Portal.
- The Oracle databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle server will be remote or local to the WebSphere Portal installation.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle database in order for WebSphere Portal to communicate with the Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes:
  
  
  - `db_block_size = 8192` bytes
  - `db_cache_size = 307,200` bytes
  - `db_files = 1024` files
  - `log_buffer = 65536` bytes
  - `open_keywords = 1500` keywords
  - `pga_aggregate_target = 204,800` bytes
  - `pre_page_sga = true`
  - `processes = 300` processes
  - `shared_pool_size = 204,800` bytes

  **Note:** If you are using IBM Java Content Repository, the `open_keywords` value may need to be increased based on the table count in the Java Content Repository schema.

- Raise the number of parallel servers as appropriate. For example, if you have more than 875 parallel servers, you should set the `parallel_max_servers` to 1200.
- The Oracle parameter `CURSOR_SHARING` allows similar SQL Statements to be shared when possible, which prevents parsing and establishing a new execution plan. The execution plan is used by Oracle to gather the data needed to satisfy a request. There are two options for `CURSOR_SHARING`, which are as follows:
  
  **FORCE**
  
  When you select this option, Oracle uses the same execution plan for all SQLs that are similar in value even if the values are different. When you use this option, the execution plan may not provide optimum performance. For example, similar SQLs with different values may behave differently when executed running the same plan.

  **EXACT**
- When you select this option, Oracle only shares the same execution plan for SQLs that are identical and use the same values. This option removes the risk of a SQL statement being executed when optimum performance conditions do not exist.

WebSphere Portal supports both options. Regardless of the option selected, portlet applications should not be affected. Contact your database administrator for further assistance on these options.

**Prerequisites**

- Installing Oracle
- Creating users

Parent topic: Configuring WebSphere Portal to use Oracle

Previous topic: Creating users

Next topic: Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- Prerequisites
  - Installing Oracle
  - Creating users
  - Creating databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

Tip: For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Working with properties files:
- The WebSphere Portal database can be used to hold information for applications such as Feedback and Like Minds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:
  - release.DbName=release
  - jcr.DbName=jcrdb
  - feedback.DbName=fdbkdb
  - likeminds.DbName=lmdb
  - community.DbName=commdb
  - customization.DbName=customdb

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds

- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
- `dbdomain DbType`
- `dbdomain DbName`
- `dbdomain DbUrl`
- `dbdomain DbSchema`

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If `DbUser`, `DbUrl`, and `DbPassword` are not the same across domains, the value for `DataSourceName` must differ from the `DataSourceName` of the other domains. In other words, this value must be unique for the database domain.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

1. Locate the following files and create a backup copy of each before changing any values:
   - `wp_profile_root/ConfigEngine/properties/wkplc.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_comp.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties`

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In `wkplc_comp.properties`, most properties are repeated for each domain.

2. Use a text editor to open the properties file `wkplc_comp.properties` and modify the values to correspond to your environment.

   A. For `dbdomain DbType`, type `oracle`.
   B. For `dbdomain DbName`, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the `dbdomain DbName` property.
   C. For `dbdomain DbSchema`, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The `dbdomain DbName` should be the same value used for the `dbdomain DbSchema` restricted.

   **Restriction:** The value for `dbdomain DbSchema` must equal the value for `dbdomain DbUser`.
   D. For `dbdomain DataSourceName`, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For `dbdomain DbUrl`, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:**
      - The database element of this value should match the value of DbName.
      - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following:
        `jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME`. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.
F. For `dbdomain.DbUser`, type the user ID for the database administrator. **Restriction:** The value for `dbdomain.DbUser` must equal the value for `dbdomain.DbSchema`.

G. For `dbdomain.DbPassword`, type the password for the database administrator.

H. For `dbdomain.DBA.DbUser`, type the database administrator user ID for privileged access operations during creation of the database.

I. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.

J. For `dbdomain.DbHome`, type the root location for the database. **Note:** This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For `source.domainDbType`, type of the database you are currently configured to use. The value for `source.domainDbType` is Derby by default.

B. For `source.domainDbName`, type the name of the database domain you are currently using.

C. For `source.domainDbSchema`, type current schema identifier for objects within the database for this domain.

D. For `source.domainDataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For `source.domainDbUrl`, type the url currently used to access your database.

F. For `source.domainDbUser`, type the name of the user accessing this database.

G. For `source.domainDbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

   A. For `oracle.dbDriver`, type the name of the class that SqlProcessor uses to import SQL files.

   B. For `oracle.dbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For `oracle.jdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Configuring WebSphere Portal to use Oracle  
**Previous topic:** Creating databases  
**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

- **Prerequisites**
  - Installing Oracle
  - Creating users
  - Creating databases
  - Modifying database properties

1. On the database server, make sure that the folder `db_1` contains the subfolders `your_oracle_instance/data` and `your_oracle_instance/index`. If this folder hierarchy does not exist, create it manually before you run the `setup-database` task. The `setup-database` task requires these folders to create database users. If these folders do not exist, the `setup-database` task will fail.

2. Change to the directory `wp_profile_root/ConfigEngine`.

3. To create the database users, type the following command:

   ```bash
   ./ConfigEngine.sh setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:

   `<WP_root>/base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql`.

---

**Parent topic:** Configuring WebSphere Portal to use Oracle  
**Previous topic:** Modifying database properties  
**Next topic:** Creating JCR table spaces

**Related tasks**  
Manually creating users and granting privileges for Oracle
Creating JCR table spaces

This topic provides manual instructions for creating JCR table spaces for Oracle.

- **Prerequisites**
  - Installing Oracle
  - Creating users
  - Creating databases
  - Modifying database properties
  - Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.
2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:
   - `&jcrdb` is the name of the database you created to store user data.
   - `&dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.

   Ensure that the `.` is included in the variables when you substitute the values of your environment with these variables.

   **Important:** You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

   ```sql
   create tablespace ICMLFQ32 datafile '/&dbpath./&jcrdb./data/icmlfq32_01.dbf' size 300M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLNF32 datafile '/&dbpath./&jcrdb./data/icmlnf32_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLVFQ04 datafile '/&dbpath./&jcrdb./data/icmlvfq04_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMSFQ04 datafile '/&dbpath./&jcrdb./data/icmsfq04_01.dbf' size 150M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLSNDX datafile '/&dbpath./&jcrdb./index/icmlsndx_01.dbf' size 10M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   ```

   A. Set the `size`, `autoextend`, and `maxsize` values according to your environment. For example, you may want to change the `maxsize` to a set value rather than `UNLIMITED`.

   B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.

   C. Refer to the Oracle command reference for more information about using the `create tablespace` command.

**Parent topic:** Configuring WebSphere Portal to use Oracle

**Previous topic:** Setting up databases

**Next topic:** Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:

- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root`/PortalServer/config/tables spaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

- Prerequisites
  - Installing Oracle
  - Creating users
  - Creating databases
  - Modifying database properties
  - Setting up databases
  - Creating JCR table spaces

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tables spaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`
   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
     - `release`
     - `community`
     - `customization`
     - `jcr`
     - `feedback`
     - `likeminds`
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword `TABLESPACE` and a space. For example: `community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE`
   Repeat this step for each domain that you are transferring.
4. Save and close `dbdomain.space_mapping.properties`.
5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - `Windows`: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

Parent topic: Configuring WebSphere Portal to use Oracle
Previous topic: Creating JCR table spaces
Next topic: Configure WebSphere Portal to use Oracle
Configure WebSphere Portal to use Oracle

This section provides information on how to manually transfer data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- Prerequisites
  - Installing Oracle
  - Creating users
  - Creating databases
  - Modifying database properties
  - Setting up databases
  - Creating JCR table spaces

Tips:
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

Steps for transferring data to another supported database

1. Open a command prompt and change to the directory \wp_profile_root\ConfigEngine.
2. Enter the following commands to validate configuration properties: 
   .\ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   .\ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
3. From the same command prompt as the previous steps, change to the directory \wp_profile_root\bin.
4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td>./stopServer.sh server1 -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>
5. **Transfer the database**:

   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   **A.** Change to the directory `wp_profile_root/ConfigEngine`.
   
   **B.** Enter the following command:
   
   ```
   ./ConfigEngine.sh database-transfer -DWasPassword=password
   ```
   
   **Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:
   
   ```
   ./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
   ```
   
   **C.** After running this task, a message is added to the log files to verify that this task was successful. Check the log files.
   
   If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example:

   ```
   SQL> execute dbms_stats.gather_schema_stats(ownname=> 'jcr', cascade=> TRUE);
   ```

7. Change to the directory `wp_profile_root/bin`.

8. Enter the following command to start the WebSphere Portal server:

   ```
   ./startServer.sh WebSphere_Portal -username admin_userid -password admin_password
   ```

---

**Parent topic:** Configuring WebSphere Portal to use Oracle  
**Previous topic:** Assigning custom table spaces
Configuring WebSphere Portal to use Oracle RAC

View information on installing and setting up Oracle RAC to work with WebSphere Portal.

1. Installing Oracle RAC
   View information on how to install Oracle RAC for use with WebSphere Portal.

2. Create users
   View the steps to set up users for Oracle RAC to work with WebSphere Portal.

3. Creating databases
   View some important considerations before setting up Oracle RAC databases to work with WebSphere Portal.

4. Modifying database properties
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

6. Creating JCR table spaces
   This topic provides manual instructions for creating JCR table spaces for Oracle.

7. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

8. Configuring WebSphere Portal to use Oracle RAC
   This section provides information on how to manually transfer data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configure WebSphere Portal to use a remote database
Installing Oracle RAC

View information on how to install Oracle RAC for use with WebSphere Portal.

Before you begin:
- You should have completed reviewing the Planning for Oracle RAC section.
- Ensure the Oracle CRS (Cluster Ready Service) and Oracle RAC databases have been installed and configured on the primary and secondary nodes.
- Run the following commands on both RAC nodes to start global services daemon_(GSD), oracle listeners, and agents.
  - `$ gsdctl start`
  - `$ lsnrctl start`
  - `$ agentctl start`

You can use Oracle RAC as the database software for WebSphere Portal. You must install Oracle RAC separately and before performing WebSphere Portal database transfer.

Refer to Oracle product documentation for installation instructions.

Parent topic: Configuring WebSphere Portal to use Oracle RAC

Next topic: Create users
Create users

View the steps to set up users for Oracle RAC to work with WebSphere Portal.

Before you begin:

- You should have completed installing Oracle RAC.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

**Prerequisites**

- Installing Oracle RAC

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Create the users in the WebSphere Portal databases
   A. Log in to the database in which you want to create the new users.
   B. Create the users that you need in the appropriate databases by entering the following command in the SQL*Plus tool, where `user_tablespace` is the default tablespace that is identified by the database administrator to store user objects and `temp_tablespace` is identified to store temporary objects:

   ```sql
   SQL> create user username identified by password
       default tablespace user_tablespace
       temporary tablespace temp_tablespace;
   ```

   **Tip:** Balance the storage of user objects among tablespaces to prevent running out of space with overuse of `user_tablespace`. Also consider increasing the size of `user_tablespace` when handling a high volume of users.

2. Log in by entering the command `sqlplus` in SQL*Plus:

3. Enter the command `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example: system/manager@wpsdb will log the administrative user system with a password of manager into the wpsdb database.

4. Create the WebSphere Portal user `dbdomain.DbUser`, where `dbdomain` is replaced by release, community, customization, or JCR. All names must be in lower case. The following code uses the default user tablespace and the default temporary tablespace. Explicitly specify these tablespaces if you do not want to use the defaults.

   ```sql
   SQL> create user releaseusr identified by password;
   SQL> create user communityusr identified by password;
   SQL> create user customizationusr identified by password;
   SQL> create user jcr identified by password;
   ```

5. Connect to the content database by entering the command `SQL> connect`.

6. Enter the following, where `username` is an existing administrative user in the database: `user-name: username/password@dbname`. For example, `system/manager@jcrdb` will log the administrative user system with a password of manager into the Oracle RAC database.

7. Create the Oracle RAC users by entering the following: When creating the `jcr` user, grant all necessary privileges. If you do not grant privileges, you will receive the error `ICMADMIN lacks CREATE SESSION Privilege logon denied`. 

---

**WebSphere Portal, Version 6.1.5**

**Operating systems:** AIX, HP-UX, i5/OS, Linux, Solaris, Windows
when you try to connect with the \texttt{jcruser}:

```
SQL> create user jcr identified by password default tablespace users temporary tablespace temp;
```

8. Connect to the Feedback database:
   A. Enter the following command:

   ```
   SQL> connect
   ```

   B. Enter \texttt{user-name: username/password@dbname} where \texttt{username} is an existing administrative user in the database.

   For example, system/manager@fdbkdb will log the administrative user system with a password of manager into the fdbkdb database.

9. Create the Feedback user:

   ```
   SQL> create user feedback identified by password default tablespace users temporary tablespace temp;
   ```

10. Connect to the LikeMinds database:

    ```
    SQL> connect
    ```

11. Enter \texttt{user-name: username/password@dbname}, where \texttt{username} is an existing administrative user in the database. For example, system/manager@lmdb will log the administrative user system with a password of manager into the lmdb database.

12. Create the LikeMinds user:

    ```
    SQL> create user lmdbusr identified by password default tablespace users temporary tablespace temp;
    ```

13. Log out of the command line tool using the command \texttt{SQL> exit}.

\textbf{Parent topic:} Configuring WebSphere Portal to use Oracle RAC  
\textbf{Previous topic:} Installing Oracle RAC  
\textbf{Next topic:} Creating databases
Creating databases

View some important considerations before setting up Oracle RAC databases to work with WebSphere Portal. When creating Oracle RAC databases for use with WebSphere Portal, you should consider the following information:

- The Oracle RAC databases must be created manually before configuring WebSphere Portal.
- All databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle RAC server will be remote or local to the WebSphere Portal installation.
- If using an earlier version of Oracle RAC (9i or earlier), ensure that Oracle RAC JVM is also installed.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle RAC database in order for WebSphere Portal to communicate with the IBM Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes are:

  - `db_block_size` = 8192 bytes
  - `db_cache_size` = 314,572,800 bytes
  - `db_files` = 1024 files
  - `log_buffer` = 65536 bytes
  - `open_cursors` = 1500 cursors
  - `pga_aggregate_target` = 209,715,200 bytes
  - `pre_page_sga` = true
  - `processes` = 300 processes
  - `shared_pool_size` = 209,715,200 bytes

  **Note:** If you are using IBM Java Content Repository, the `open_cursors` value may need to be increased based on the table count in the IBM Java Content Repository schema.

**Prerequisites**

- Installing Oracle RAC
- Create users

Refer to the following instructions to create tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.
2. Create the tablespaces. **Important:** You must use the same tablespace names listed in the commands. The tablespace names cannot be customized or modified.

   A. Find and edit the SQL script `jcr_ora_tablespaces.sql` in the directory `/ConfigEngine/work/db/oracle`.
   
   B. In the `define` section, replace the following variables with the values from your environment:
      - `jcrdb`
        - Name of the database you created to store user data.
      - `logfile`
        - Location to store the log file.
      - `dbpath`
        - Directory where you created the database.
   
   C. Set the size, autoextend, and maxsize values according to your environment. For example, you may want to change the maxsize to a set value rather than UNLIMITED. Consult your Database Administrator for more info.
D. Execute the following SQL script:

- The DBA or a user with sufficient credentials (CREATE tablespace) must execute the script.
- The script will prompt for the database username and password.

```
# sqlplus
SQL > @jcr_ora_tablespaces.sql
```

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC  
**Previous topic:** Create users  
**Next topic:** Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- Prerequisites
  - Installing Oracle RAC
  - Create users
  - Creating databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

TIP: For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Working with properties files:
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:
  - release.DbName=release
  - jcr.DbName=jcrdb
  - feedback.DbName=fdbkdb
  - likeminds.DbName=lmdb
  - community.DbName=commdb
  - customization.DbName=custdb

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds

- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
1. If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.

   A. For dbdomainDbType, type oracle.
   B. For dbdomainDbName, type the name of the WebSphere Portal domain database. Note: This value is also the database element in the dbdomainDbUrl property.
   C. For dbdomainDbSchema, type the schema name of the database domain. Note: Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The dbdomainDbName should be the same value used for the dbdomainDbSchema
      Restriction: The value for dbdomainDbSchema must equal the value for dbdomainDbUser.
   D. For dbdomainDataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For dbdomainDbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. Note: The database element of this value should match the value of DbName.
      - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following:
        jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.
F. For `dbdomain.DbUser`, type the user ID for the database administrator. **Restriction:** The value for `dbdomain.DbUser` must equal the value for `dbdomain.DbSchema`.

G. For `dbdomain.DbPassword`, type the password for the database administrator.

H. For `dbdomain.DBA.DbUser`, type the database administrator user ID for privileged access operations during creation of the database.

I. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.

J. For `dbdomain.DbHome`, type the root location for the database. **Note:** This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For `source.domainDbType`, type of the database you are currently configured to use. The value for `source.domainDbType` is Derby by default.

B. For `source.domainDbName`, type the name of the database domain you are currently using.

C. For `source.domainDbSchema`, type current schema identifier for objects within the database for this domain.

D. For `source.domainDataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For `source.domainDbUrl`, type the url currently used to access your database.

F. For `source.domainDbUser`, type the name of the user accessing this database.

G. For `source.domainDbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

A. For `oracle.DbDriver`, type the name of the class that `SqlProcessor` uses to import SQL files.

B. For `oracle.DbLibrary`, type the directory and name of the `.zip` or `.jar` file that contains the JDBC driver class.

C. For `oracle.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC  
**Previous topic:** Creating databases  
**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

Before you begin, ensure that the following prerequisites are met:
- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

**Prerequisites**
- Installing Oracle RAC
- Create users
- Creating databases
- Modifying database properties

1. On the database server, make sure that the folder db_1 contains the subfolders your_oracle_instance/data and your_oracle_instance/index. If this folder hierarchy does not exist, create it manually before you run the setup-database task. The setup-database task requires these folders to create database users. If these folders do not exist, the setup-database task will fail.

2. Change to the directory wp_profile_root/ConfigEngine

3. To create the database users, type the following command:

   ```
   ./ConfigEngine.sh setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires db_owner permission for the database user. The script that creates users is invoked by the configuration task setup-database and is located in the installation directory of WebSphere Portal:

   `<WP_root>/base/wp.db.impl/config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql`.

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC

**Previous topic:** Modifying database properties

**Next topic:** Creating JCR table spaces

**Related tasks**
Manually creating users and granting privileges for Oracle
Creating JCR table spaces

This topic provides manual instructions for creating JCR table spaces for Oracle.

- **Prerequisites**
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties
  - Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.
2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:
   - `&jcrdb` is the name of the database you created to store user data.
   - `&dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.

   Ensure that the `'.'` is included in the variables when you substitute the values of your environment with these variables.

   **Important:** You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

   ```sql
   create tablespace ICMLFQ32 datafile '&dbpath./&jcrdb./data/ICMLFQ32_01.dbf' size 300M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLNF32 datafile '&dbpath./&jcrdb./data/ICMLNF32_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMVFQ04 datafile '&dbpath./&jcrdb./data/ICMVFPQ04_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMSFQ04 datafile '&dbpath./&jcrdb./data/ICMSFPQ04_01.dbf' size 150M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLSNDX datafile '&dbpath./&jcrdb./index/ICMLSNDX_01.dbf' size 10M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   
   A. Set the `size`, `autoextend`, and `maxsize` values according to your environment. For example, you may want to change the `maxsize` to a set value rather than `UNLIMITED`.
   B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.
   C. Refer to the Oracle command reference for more information about using the `create tablespace` command.

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC  
**Previous topic:** Setting up databases  
**Next topic:** Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

**Prerequisites**
- Installing Oracle RAC
- Create users
- Creating databases
- Modifying database properties
- Setting up databases
- Creating JCR table spaces

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`
   
   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds

3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword `TABLESPACE` and a space. For example: `community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE`

   Repeat this step for each domain that you are transferring.

4. Save and close `dbdomain.space_mapping.properties`.

5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - Windows: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
UNIX: /ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

Parent topic: Configuring WebSphere Portal to use Oracle RAC
Previous topic: Creating JCR table spaces
Next topic: Configuring WebSphere Portal to use Oracle RAC
Configuring WebSphere Portal to use Oracle RAC

This section provides information on how to manually transfer data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

**Prerequisites**
- Installing Oracle RAC
- Create users
- Creating databases
- Modifying database properties
- Setting up databases
- Creating JCR table spaces

**Tips:**
- If you are transferring from Oracle, the `open_cursors` setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

- The WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following: `jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME`. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.
- Manually specify an RAC datasource URL in the WebSphere Application Server console. The URL should look like the following:
  
  `jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP) (HOST=PRIMARY_NODE_HOSTNAME) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP) (HOST=SECONDARY_NODE_HOSTNAME) (PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICE))`.  

- When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

**Steps for transferring data to another supported database**
1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.
2. Enter the following commands to validate configuration properties:
   ```bash
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```
3. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.
4. Stop both WebSphere Application Server and the WebSphere Portal server:
   ```bash
   ./stopServer.sh server1 -username admin_userid -password admin_password
   ./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password
   ```
5. Transfer the database:**Note**: Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.
   A. Change to the directory `wp_profile_root/ConfigEngine`.
   B. Enter the following command:
      ```bash
      ./ConfigEngine.sh database-transfer -DWasPassword=password
      ```
      **Note**: To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:
      ```bash
      ./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
      ```
   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.
6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example:
   ```sql
   SQL> execute dbms_stats.gather_schema_stats(ownname=> 'jcr', cascade=> TRUE);
   ```
7. Specify the JDBC URL to connect to the cluster:
   A. Login to the WebSphere Application Server Administrator Console
   B. Navigate to **Resources > JDBC Providers**
   C. If there is a value in the Node field, remove it and click **Apply**.
   D. For each Oracle JDBC provider, repeat the following steps:
      1. Click the provider name.
      2. Select **Data Sources**.
      3. Click the name of the data source.
      4. Under the Oracle data source properties, modify the URL attribute according to Oracle RAC configuration:
         ```sql
         jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICENAME)))
         ```
      5. Save your changes
8. Change to the directory `wp_profile_root/bin`.
9. Enter the following command to start the WebSphere Portal server:
   ```bash
   ./startServer.sh WebSphere_Portal
   ```

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC

**Previous topic:** Assigning custom table spaces
Configuring WebSphere Portal to use SQL Server

View information on installing and setting up SQL Server to work with WebSphere Portal.

1. Installing SQL Server
   View the steps to install SQL Server for use with WebSphere Portal.

2. Modifying database properties
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc DbType.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

3. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

4. Optional: Assigning custom filegroups
   The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

5. Configuring WebSphere Portal to use SQL Server 2005
   This section provides information on how to manually transfer data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configure WebSphere Portal to use a remote database
Installing SQL Server

View the steps to install SQL Server for use with WebSphere Portal. Before you begin this task, complete the following prerequisites:

- You should have completed reviewing the Planning for SQL Server topic.
- You must install SQL Server separately from WebSphere Portal.
- WebSphere Portal requires either the DataDirect Connect for JDBC 3.7 JDBC driver or the Microsoft SQL Server JDBC Driver.
- You can obtain the DataDirect Driver from DataDirect Technologies. The JDBC driver consists of two parts: the JDBC driver libraries and JTA libraries. The JDBC driver libraries must be put on the WebSphere Portal system, while the JTA libraries are intended for SQL Server.
- You can obtain the Microsoft SQL Server JDBC Driver from Microsoft. See the Microsoft SQL Server product documentation for installation details.

The following section provides instructions for installing SQL Server for use with IBM® WebSphere® Application Server and WebSphere Portal. These steps are the same for both the DataDirect and Microsoft drivers unless noted.

1. Install SQL Server and all required patches.
2. Select the authentication mode for this installation:
   - Windows Authentication Mode
   - Mixed Mode (Windows Authentication and SQL Server Authentication)
     Either authentication mode allows the user who is specified as the database administrator in the wkplc_comp.properties file to log in to the SQL server. Mixed Mode authentication allows either a Windows user or a SQL server user, or both, to log in to the SQL Server.
3. In the SQL Server Setup panel, Components to Install, select the following components, which are required services for WebSphere Portal:
   - SQL Server Database Services
     - Integration Services The option Integration Services creates the directory DTS/Binn, where you will need to copy files when you install Microsoft SQL Server JDBC drivers and enable XA connections.
4. Complete the installation using SQL Server documentation as a guide.
5. Enable TCP/IP connectivity in the SQL Server Configuration Manager.
6. Install the JDBC driver using one of these methods:
   - Installing DataDirect Connect for JDBC drivers on UNIX
   - Installing DataDirect Connect for JDBC drivers on Windows
   - Installing Microsoft SQL Server JDBC drivers and enabling XA connections
7. Network the connections for MSDTC in the Windows Component Services as described in Microsoft Help and Support.
8. Start SQL Server.

Installing DataDirect Connect for JDBC drivers on UNIX

1. Purchase and download DataDirect Connect for JDBC and save file 360connectjdbc.jar in a temporary work directory.
To create the required files, run the following command from the directory that contains 360connectjdbc.jar:

```
jar -xvf 360connectjdbc.jar
```

Run `./installer.sh` in the same directory.

When installation is complete, change directory to `/installation/lib`, where `installation` is the directory in which you installed the DataDirect Connect for JDBC drivers.

Run the following command to change the permissions on the JAR files in the `installation/lib` directory:

```
chmod 777 *.jar
```

Change the ownership and group of these files by running the following commands in the `installation/lib` directory:

```
chgrp system_grp *.jar
chown root *.jar
```

Where `system_grp` is the system group as labeled by your operating system.

Installing Microsoft SQL Server JDBC drivers and enabling XA connections

1. Download and install the Microsoft SQL Server JDBC driver.

2. Copy file `sqljdbc_xa.dll` from the `xa` subdirectory to the `C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqljdbc_xa.dll` directory of the SQL Server installation.

3. Start the database server.

4. Ensure that the Distributed Transaction Coordinator has been started. The status can be verified in the list of services in the Computer Management console.

5. Start the Microsoft SQL Server Management Studio and connect to the local database engine as the system administrator, `sa`.

6. Select File > Open > File and select `xa_install.sql` from the subdirectory of the downloaded and extracted JDBC driver.

7. Execute the script by selecting Query > Execute. Note: Any warnings that appear in the messages section of the application window that say that stored procedures cannot be found can be safely ignored.

8. For Microsoft SQL Server JDBC drivers: If you are running Windows XP SP2, Windows XP 64-Bit Edition, or Windows Server 2003, refer to the Registry Entries Are Required for XA Transaction Support document for information on a new security constraint and how to set SQL Server on Windows XP SP2, Windows XP 64-Bit Edition, or Windows Server 2003. Create an additional value in the Windows registry for WebSphere Portal by following these steps:

   A. Open the Windows Registry Editor (regedit) and navigate to the element `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSDTC\XA\DLL`

   B. From the menu bar, select Edit > New > String Value to create a new parameter named `sqljdbc_xa.dll` in that element.

   C. Change the value of the new parameter to the location of the `sqljdbc_xa.dll` file copied in Step 2 above, for example: `C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqljdbc_xa.dll`

Parent topic: Configuring WebSphere Portal to use SQL Server

Next topic: Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

**Prerequisites**

- Installing SQL Server

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:

- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**

- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:

  - `release.DbName=release`
  - `jcr.DbName=jcrdb`
  - `feedback.DbName=fdbkidb`
  - `likeminds.DbName=lmdb`
  - `community.DbName=commdb`
  - `customization.DbName=custdb`

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace `dbdomain` with:

  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`

- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:

  - `dbdomain.DbType`
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If \texttt{DbUser}, \texttt{DbUrl}, and \texttt{DbPassword} are not the same across domains, the value for \texttt{DataSourceName} must differ from the \texttt{DataSourceName} of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
\begin{itemize}
  \item \texttt{wp_profile_root}/ConfigEngine/properties/wkplc.properties
  \item \texttt{wp_profile_root}/ConfigEngine/properties/wkplc_comp.properties
  \item \texttt{wp_profile_root}/ConfigEngine/properties/wkplc_dbtype.properties
\end{itemize}
Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In \texttt{wkplc_comp.properties}, most properties are repeated for each domain.

2. Use a text editor to open the properties file \texttt{wkplc_comp.properties} and modify the values to correspond to your environment.
\begin{itemize}
  \item A. For \texttt{dbdomainDbType}, type \texttt{sqlserver2005}.
  \item B. For \texttt{dbdomainDbName}, type the name of the WebSphere Portal domain database. \textbf{Note:} This value is also the database element in the \texttt{dbdomainDbUrl} property.
  \item C. For \texttt{dbdomainDbSchema}, type the schema name of the database domain. \textbf{Note:} Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
  \item D. For \texttt{dbdomainDataSourceName}, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
    \begin{itemize}
      \item releaseDS
      \item communityDS
      \item customizationDS
      \item jcrDS
      \item lmdbDS
      \item feedback
    \end{itemize}
  \item E. For \texttt{dbdomainDbUrl}, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. \textbf{Note:} The database element of this value should match the value of \texttt{DbName}.
  \item F. For \texttt{dbdomainDbUser}, type the user ID for the database administrator.
  \item G. For \texttt{dbdomainDbPassword}, type the password for the database administrator.
  \item H. For \texttt{dbdomainDBA_DbUser}, type the database administrator user ID for privileged access operations during creation of the database.
  \item I. For \texttt{dbdomainDBA_DbPassword}, type the database administrator password for privileged access operations during creation of the database.
  \item J. For \texttt{dbdomainDbHome}, type the root location for the database. \textbf{Note:} This value is the location to store the database files locally.
  \item K. For \texttt{dbdomainAdminUrl}, type the sqlserver URL without a database attached. This value is used to connect to the server for database administration operations.
  \item L. For \texttt{dbdomainDbHostName}, type the hostname of the database.
\end{itemize}

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in \texttt{wkplc_comp.properties}. \textbf{Important:} The default values for the following parameters are
set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For `source.domainDbType`, type of the database you are currently configured to use. The value for `source.domainDbType` is Derby by default.

B. For `source.domainDbName`, type the name of the database domain you are currently using.

C. For `source.domainDbSchema`, type current schema identifier for objects within the database for this domain.

D. For `source.domainDataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For `source.domainDbUrl`, type the url currently used to access your database.

F. For `source.domainDbUser`, type the name of the user accessing this database.

G. For `source.domainDbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

   A. For `sqlserver2005.DbDriver`, type the name of the class that `SqlProcessor` uses to import SQL files.

   B. For `sqlserver2005.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For `sqlserver2005.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

   D. For `sqlserver2005.DbConnectionPoolDataSource`, type the name of the implementation class of the connection pool data source.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Configuring WebSphere Portal to use SQL Server  
**Previous topic:** Installing SQL Server  
**Next topic:** Setting up databases
Setting up databases
This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

Before you begin, ensure that the following prerequisites are met:
- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

- Prerequisites
  - Installing SQL Server
  - Modifying database properties

1. Change to the directory `wp_profile_root/ConfigEngine`
2. To create the databases, type the following command:
   ```bash
   ./ConfigEngine.sh create-database -DWasPassword=password
   ```
3. To create the database users, type the following command:
   ```bash
   ./ConfigEngine.sh setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:
   `<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql`

Parent topic: Configuring WebSphere Portal to use SQL Server
Previous topic: Modifying database properties
Next topic: Assigning custom filegroups

Related tasks
Create users and databases for SQL Server 2005 on AIX and UNIX
Assigning custom filegroups

The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

Before you begin:

- The custom filegroups must exist prior to the execution of the database transfer.
- To see which table spaces can be customized in each domain, refer to the wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating filegroups refer to the documentation for the database.

**Prerequisites**

- Installing SQL Server
- Modifying database properties
- Setting up databases

If custom filegroups are assigned, each must be assigned explicitly. The default filegroups can be used to contain database objects; however the name of the default file group must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom filegroups:

1. Determine the names of your custom filegroups.
2. Open the mapping file wp_profile_root/PortalServer/config/tablespaces that specifies the table space and index space for each property pairs for each database table:
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.indexspace.indexspace
   For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds
3. Assign a filespace to each entry in the mapping file. The filegroup name must be prepended by the keyword ON and a space. For example: community.COMP_INST.tablespace=ON COMM8KSPACE
   Repeat this step for each domain that you are transferring.
4. Save and close dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
   - UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
   - i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
Configuring WebSphere Portal to use SQL Server 2005

This section provides information on how to manually transfer data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005.

As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- Prerequisites
  - Installing SQL Server
  - Modifying database properties
  - Setting up databases

Tips:
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

Steps for transferring data to another supported database

1. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.
2. Enter the following commands to validate configuration properties.
   ```
   ../ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ../ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```
3. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.
4. Stop both WebSphere Application Server and the WebSphere Portal server:
   ```
   Option                      Description
   WebSphere Application Server ./stopServer.sh server1 -username admin_userid -password admin_password
   WebSphere Portal            ./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password
   ```
5. Transfer the database: Note: Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.
   A. Change to the directory wp_profile_root/ConfigEngine.
B. Enter the following commands:
```
./ConfigEngine.sh database-transfer -DWasPassword=password
```

**Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:
```
./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
```

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. Change to the directory `wp_profile_root/bin`.

7. Enter the following command to start the WebSphere Portal server:
```
./startServer.sh WebSphere_Portal
```

8. Update the SQL Server 2005 statistics for Portal, and JCR databases by opening SQL Server Management Studio, selecting **New Query**, and running the following query:
```
use db_name
exec sp_updatestats @resample='resample';
```

**Parent topic:** Configuring WebSphere Portal to use SQL Server

**Previous topic:** Assigning custom filegroups
Verifying databases
After you configure IBM® WebSphere® Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.

- **Prerequisites**
  - [+] Technotes for database connectivity issues

You can verify the database connection using IBM WebSphere Application Server or by opening WebSphere Portal in a browser.

- To verify that the WebSphere Portal application server is running by using WebSphere Application Server, complete these steps:
  1. Open the WebSphere Application Server administrative console by entering the following address in a browser:
     ```
     http://hostname.example.com:10027/ibm/console
     ```
     where `hostname.example.com` is the fully qualified host name of the machine where WebSphere Portal is running and `10027` is the default transport port that is created by WebSphere Application Server.
  2. Log into the administrative console.
  3. Depending on your version of WebSphere Application Server, click the appropriate option:
     - For **WebSphere Application Server Version 6.1**: Click Resources & JDBC Providers.
     - For **WebSphere Application Server Version 7.0**: Click Resources > JDBC > JDBC Providers
  4. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
  5. Select the name of the data source that is defined in `wkplc_comp.properties`. The default data source is `wpdbDS`.
  6. Select the name of the JDBC provider that is specified in `wkplc_dbtype.properties`. The default JDBC provider is `wpdbJDBC_dbtype`, where `dbtype` is replaced by the value that matches your environment.
  7. Click **Test Connection** to verify the database connection. If configuration parameters have been changed, you might need to restart WebSphere Application Server for the test to complete.

- To verify that the WebSphere Portal application server is running by opening WebSphere Portal in a browser, enter the following URL in a supported browser:
  ```
  http://hostname.example.com:10040/wps/portal
  ```
  where `hostname.example.com` is the fully qualified host name of the machine where WebSphere Portal is running and `10040` is the default transport port that is created by WebSphere Application Server.

**Parent topic:** Configure WebSphere Portal to use a remote database
Configuring the primary node to communicate with the deployment manager on HP-UX

After installing IBM® WebSphere® Application Server Network Deployment and installing IBM WebSphere Portal on the primary node, you must configure the primary node and the deployment manager to communicate with each other.

- Prerequisites
  - Installing WebSphere Portal on HP-UX on the primary node
  - Configure WebSphere Portal to use a remote database

Perform the following steps to configure WebSphere Portal to communicate with the deployment manager:

1. Run the `./ConfigEngine.sh collect-files-for-dmgr -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory of the primary node, to create an archive or compressed file containing all the files which need to be copied to the Deployment Manager. **Note:** The archive or compressed file will be placed in the `wp_profile_root/filesForDmgr` directory and the file will be called `filesForDmgr.zip`.

2. Stop the deployment manager.

3. Expand the `filesForDmgr.zip` file into the installation root directory of the Deployment Manager; for example in the `/opt/IBM/WebSphere/AppServer` directory. **Note:** If the Deployment Manager profile was not created in the default `AppServer/profiles/Dmgr01` directory, then the `metadata_wkplc.xml` file, located in the `AppServer/profiles/Dmgr01/config/.repository/metadata_wkplc.xml` directory in the zip file, must be placed into the correct Deployment Manager profile directory.

4. If the Deployment Manager profile is running on the same application server where WebSphere Portal is installed, remove the `com.ibm.ws.portletcontainer.deploytask_6.1.5.jar` file from the `AppServer/plugins` directory.

5. Start the deployment manager.

**Parent topic:** Preparing the primary node on HP-UX  
**Previous topic:** Configure WebSphere Portal to use a remote database  
**Next topic:** Removing search collections on HP-UX

**Related tasks**
- Configuring Portal Search in a cluster on AIX
Removing search collections on HP-UX

If you plan to use search in a cluster, you must configure a remote search server; see Configuring Portal Search in a cluster for information. If you created any search collections, you will need to recreate them on the remote search server. If your search collection has data, export the collection before deleting it and then import the data on the remote server.

**Prerequisites**
- Installing WebSphere Portal on HP-UX on the primary node
- Configure WebSphere Portal to use a remote database
- Configuring the primary node to communicate with the deployment manager on HP-UX

Perform the following step to remove the search collection:

Perform the following steps to delete all existing search collections from the primary node:

1. Log on to WebSphere Portal.
2. Navigate to Administration > Search Administration > Manage Search and then click Search Collections.
3. Click the Delete Collection icon for each search collection and then click OK until they are all deleted.
4. Restart the WebSphere_Portal server and then navigate back to the Search Collections page to verify that all search collections have been deleted.

Parent topic: Preparing the primary node on HP-UX

Previous topic: Configuring the primary node to communicate with the deployment manager on HP-UX
Choosing the type of cluster to create on HP-UX

If you installed a IBM® WebSphere® Application Server Network Deployment, create a static cluster to handle failover requests. If you installed a IBM WebSphere Virtual Enterprise, create a dynamic cluster to balance member workloads.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on HP-UX
  - Preparing your HP-UX operating system
  - Preparing the primary node on HP-UX

Choose one of the following options to create a cluster:

- **Creating a static cluster on HP-UX**
  After installing IBM WebSphere Portal on the primary node, configuring a remote database, and preparing the primary node to communicate with the Deployment Manager, you can create your static cluster to handle failover requests.

- **Creating a new dynamic cluster on HP-UX using WebSphere Virtual Enterprise**
  After installing and configuring IBM WebSphere Portal on your primary node and all additional nodes, you can create a new dynamic cluster. A dynamic cluster monitors performance and load information and is able to dynamically create and remove cluster members based on the workload.

**Parent topic:** Setting up a cluster on HP-UX  
**Previous topic:** Preparing the primary node on HP-UX  
**Next topic:** Preparing a remote Web server when portal is installed on HP-UX
Creating a static cluster on HP-UX

After installing IBM® WebSphere® Portal on the primary node, configuring a remote database, and preparing the primary node to communicate with the Deployment Manager, you can create your static cluster to handle failover requests.

Perform the following steps to create the cluster:

1. If the Deployment Manager is configured to use a stand-alone LDAP user registry, update the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory, on the primary node with the stand-alone LDAP user registry property values from the Deployment Manager. You can find these settings under the VMM Stand-alone LDAP configuration heading.
   
   **Important:** Ensure that you set WasUserid and WasPassword to the Deployment Manager user ID and password.

2. Stop the server1 and WebSphere_Portal servers on the primary node and ensure that the following parameters are set correctly in the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory:
   
   **Note:** Although you can add these parameters directly to any task that you run while creating your cluster, you may want to add them to the properties file while creating your cluster and then remove them when you are finished to keep your environment secure.
   
   A. Set WasSoapPort to the port used to connect remotely to the deployment manager.
   B. Set WasRemoteHostName to the full host name of the server used to remotely connect to the deployment manager.
   C. Verify that WasPassword is set to your deployment manager password.
   D. Verify that PortalAdminPwd is set to your WebSphere Portal password.
   E. Verify that ClusterName is set.
   F. Verify that PrimaryNode is set to true.

3. Make a backup copy of the wp_profile_root/config/cells/cell_name/wim/config/wimconfig.xml and wp_profile_root/config/cells/cell_name/wim/model/wimxmlextension.xml, if available, files.

4. Run the ./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=dmgr_password task, from the wp_profile_root/ConfigEngine directory of the primary node.
   
   **Note:** If you want to specify custom ports for the nodeagent, add the -DPortPropsFile=full to portsfile parameter to the cluster-node-config-pre-federation task. You can use the ports files that are found on the Setup CD for WebSphere_Portal and server1 as a guide.

   **Note:** You may receive a message about accepting an SSL signer certificate. Failure to accept the SSL signer certificate will cause the script to fail. Alternatively, the com.ibm.ssl.enableSignerExchangePrompt flag can be enabled in the ssl.client.props file for "DefaultSSLSettings" in order to allow acceptance of the signer during the connection attempt.

   **Warning:** If the cluster-node-config-pre-federation fails for any reason, you must perform the following steps before rerunning the task:
   
   A. Remove the node if the AddNode task succeeded.
   B. Log on to the deployment manager and perform the following steps if the items exist:
      1. Remove all enterprise applications.
      2. Remove the WebSphere_Portal server definition.
      3. Remove the WebSphere Portal JDBC Provider.

   **Note:** If you are migrating from a version prior to Version 6.1.0 that contains Lookaside data, read the following technote: Migrating with Lookaside Data.
5. Optional: If you configured a database user registry or a property extension (lookaside) database, run the following task to set up access to the database drivers:

**Note:** You will also need to run this task if you migrated the primary node from a release prior to Version 6.1.0, even if you did not configure a database user registry or a property extension (lookaside) database.

A. Set the property value for `federated.db.DbType` if using a database user registry or set the property value for `la DbType` if using a property extension database in the `wkplc.properties` file.

**Note:** If you migrated the primary node from a version prior to Version 6.1.0 and you are not using a database user registry or a property extension database, set the property value for `federated.db DbType` to the same value that is in the `wkplc.properties` file on the primary node.

B. Run the `./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=dmgr_password -DDbDomain=la|federated.db -DvmmNodeName=node_name -Db_type.NodeDbLibrary=local full path of the database jars` task from the `wp_profile_root/ConfigEngine` directory to create the variable used to access the VMM database jars.

**Note:** `VmmNodeName` is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The `db_type` in `Db_type.NodeDbLibrary` should be set to the type of database you are using, for example `db2`.

Run the `./ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=dmgr_password` task.

6. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell's user registry. Run the `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task, from the `wp_profile_root/ConfigEngine` directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

**Attention:** The password value for `-DWasPassword` is the Deployment Manager administrative password.

**Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

**Fast path:** If the value for `newAdminGroupId` contains a space; for example `Software Group`, open the `wkplc.properties` file and add the values for `newAdminId`, `newAdminPw`, and `newAdminGroupId`. Save your changes and run the `ConfigEngine.bat wp-change-portal-admin-user -DWasPassword=dmgr_password` task.

**Important:** If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user task until after the cluster-node-config-cluster-setup (static cluster) or cluster-node-config-dynamic-cluster-setup (dynamic cluster) task completes. After running the `wp-change-portal-admin-user task, start or restart the WebSphere_Portal server to use the updated administrator user ID.

**Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the `wp-change-portal-admin-user task. `-DnewAdminPw` is an optional parameter to update the Administrative password in the `wkplc.properties` file if required.

Run the `./ConfigEngine.sh cluster-node-config-cluster-setup -DWasPassword=dmgr_password` task.

8. Configure the cluster to use an external Web server to take advantage of features such as workload management. Choose one of the following options:

- Configuring a Web server and an application server on separate machines (remote)
- Configuring a Web server and an application server profile on the same machine
- Configuring a Web server and a deployment manager profile on the same machine

**Note:** Start with the step about launching the Plug-ins installation wizard.

9. Perform the following steps to access the Web Content Management content through an external Web server:

A. Log on to the deployment manager administrative console.
B. Select Environment > WebSphere Variables.

C. From the Scope drop-down menu, select the Node=nodename, Server=servername option to narrow the scope of the listed variables, where Node=nodename is the node that contains the application server.

D. Update the WCM_HOST variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.

E. Update the WCM_PORT variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.

F. Perform the following steps to synchronize the node with the deployment manager:
   1. Select System Administration > Nodes.
   2. Select the node that you want to synchronize from the list.
   3. Click Full Resynchronize.

G. Log off of the deployment manager administrative console.

11. Run the following tasks to propagate your changes: Note: WebSphere_Portal_nodename is the name of the node’s WebSphere Portal server; but if you customized the server name, the default name changes. If you are uncertain of the server name, run the serverstatus -all task to get a list of the server names and their status.

   A. ./stopManager.sh -username admin_userid -password admin_password, from the dmgr_profile_root/bin directory

   B. ./stopNode.sh -username admin_userid -password admin_password, from the wp_profile_root/bin directory

   C. ./stopServer.sh WebSphere_Portal_nodename -username admin_userid -password admin_password

   D. ./startManager.sh, from the dmgr_profile_root/bin directory

   E. ./startNode.sh, from the wp_profile_root/bin directory

   F. ./startServer.sh WebSphere_Portal_nodename

12. Change the task list settings to point to the cluster instead of a single server if using Portal business process integration. See one of the following sections in the Information Center for information:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross cell</td>
<td>Adding a BPI-enabled portal server to a managed cell in a cross-cell setup</td>
</tr>
<tr>
<td>Single cell</td>
<td>Adding a BPI-enabled portal server to a managed cell in a single-cell setup</td>
</tr>
</tbody>
</table>

**Parent topic:** Choosing the type of cluster to create on HP-UX

**Related tasks**

- Adding a BPI-enabled portal server to a managed cell in a cross-cell setup
- Adding a BPI-enabled portal server to a managed cell in a single-cell setup
Creating a new dynamic cluster on HP-UX using WebSphere Virtual Enterprise

After installing and configuring IBM® WebSphere® Portal on your primary node and all additional nodes, you can create a new dynamic cluster. A dynamic cluster monitors performance and load information and is able to dynamically create and remove cluster members based on the workload.

Before creating your dynamic cluster, install and configure the Deployment Manager and perform all the tasks under “Preparing the primary node.” On the Deployment Manager system, install WebSphere Virtual Enterprise and augment the deployment manager profile. On all WebSphere Portal nodes, install WebSphere Virtual Enterprise and augment the wp_profile profile. See Installing and configuring the product for information about installing WebSphere Virtual Enterprise. See Augmenting profiles for information about augmenting the Deployment Manager and wp_profile profiles.

- Prerequisites
  - PK97393 for all WebSphere Application Server Versions prior to 6.1.0.5
  - Installing and configuring the product:
  - Augmenting profiles

Perform the following steps to create a dynamic cluster:

1. Perform the following steps for on the primary node of the dynamic cluster:
   A. Prepare the node for the dynamic cluster; perform all tasks under Preparing the primary node.
   B. If the Deployment Manager is configured to use a stand-alone LDAP user registry, update the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory, on the primary node with the stand-alone LDAP user registry property values from the Deployment Manager. You can find these settings under the VMM Stand-alone LDAP configuration heading.
   C. Stop the server1 and WebSphere_Portal servers on the primary node and ensure that the following parameters are set correctly in the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory:
      1. Set WasSoapPort to the port used to connect remotely to the deployment manager.
      2. Set WasRemoteHostName to the full host name of the server used to remotely connect to the deployment manager.
      3. Verify that WasPassword is set to your deployment manager password.
      4. Verify that PortalAdminPwd is set to your WebSphere Portal password.
      5. Verify that ClusterName is set.
      6. Verify that PrimaryNode is set to true.
   D. Run the ./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=dmgr_password task, from the wp_profile_root/ConfigEngine directory of the primary node.

   Warning: If the cluster-node-config-pre-federation fails for any reason, you must perform the following steps before rerunning the task:
   1. Remove the node if the AddNode task succeeded.
   2. Log on to the deployment manager and perform the following steps if the items exist:
      A. Remove all enterprise applications.
      B. Remove the WebSphere_Portal server definition.
      C. Remove the WebSphere Portal JDBC Provider.

   Note: If you are migrating from a version prior to Version 6.1.0 that contains Lookaside data, read the following
technote: Migrating with Lookaside Data.

E. Run the following task. Include each node name as a comma separated list in the command:

1. Set the property value for `federated.db.DbType` if using a database user registry or set the property value for `la.DbType` if using a property extension database in the `wkplc.properties` file. **Note:** If you migrated the primary node from a version prior to Version 6.1.0 and you are not using a database user registry or a property extension database, set the property value for `federated.db DbType` to the same value that is in the `wkplc.properties` file on the primary node.

2. Run the `./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=dmgr_password -DDomain=la federated.db -DVmmNodeName=node_name -Db_type.NodeDbLibrary=localhost full path of the database jars` task from the `wp_profile_root/ConfigEngine` directory to create the variable used to access the VMM database jars. **Note:** `VmmNodeName` is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The `db_type` in `db_type.NodeDbLibrary` should be set to the type of database you are using, for example `db2`.

F. Run the `./ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=dmgr_password` task.

G. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell's user registry. Run the `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task, from the `wp_profile_root/ConfigEngine` directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

**Important:** If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user` task until after the `cluster-node-config-dynamic-cluster-setup` task completes. After running the `wp-change-portal-admin-user` task, start or restart the WebSphere_Portal server to use the updated administrator user ID.

**Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the `wp-change-portal-admin-user` task. `-DnewAdminPw` is an optional parameter to update the Administrative password in the `wkplc.properties` file if required.

2. Log on to the deployment manager administrative console.

3. Perform the following steps to create a node group:
   
   A. Click **System administration > Node groups.**
   B. Click **New.**
   C. Type the node group **Name.**
   D. Optional: Type any information about the node group in the **Description** text box.
   E. Click **OK.**
   F. Click the **Save** link to save your changes to the master configuration.

4. Perform the following steps to add members to the node group:
   
   A. Click **System administration > Node groups.**
   B. Click on the name of the node group that you want to add members to.
   C. Click **Node group members** under Additional Properties.
   D. Click **Add.**
   E. Select the primary node and then click **Add.**
   F. Click the **Save** link to save your changes to the master configuration.

5. Perform the following steps to create a dynamic cluster in the node group:
   
   A. Click **Servers > Dynamic clusters.**
   B. Click **New.**
C. Select WebSphere Application Server from the Server Type pull-down menu and then click Next.

D. Type the cluster name in the Dynamic cluster name text box and then click Next. Type the same value that you provided for the ClusterName parameter in the wkplc.properties file of your primary node.

E. Remove all default membership policies and then click Subexpression builder.

F. Enter the following information in the Subexpression builder window:

1. Select and from the Logical operator pull-down menu.
2. Select Nodegroup from the Select operand pull-down menu.
3. Select Equals (=) from the Operator pull-down menu.
4. Type the nodegroup name you created in the previous step in the Value text box.
5. Click Generate subexpression.
6. Click Append.

G. Click Preview membership to verify that all nodes included in the nodegroup display and then click Next.

H. Click the Create the cluster member using an existing server as a template radio button and then select the WebSphere_Portal server for the primary node from the pull-down menu.

I. Click Next.

J. Specify the dynamic cluster properties and then click Next.

K. Review the summary page to verify your actions and then click Finish.

L. Click the Save link to save your changes to the master configuration.

6. Define or verify the following parameters in the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory:

A. Set ClusterName to the name of the new dynamic cluster.
B. Verify that CellName is set to the deployment manager cell.
C. Verify that NodeName is set to the local WebSphere Portal node.
D. Set ServerName to the server that will be used for the dynamic cluster member on this node.

   Note: Log on to the deployment manager administrative console and click Dynamic Clusters > PortalCluster > Dynamic cluster members to find the name of the server used for the dynamic cluster.

E. Verify that PrimaryNode is set to true.

7. Run the ./ConfigEngine.sh cluster-node-config-dynamic-cluster-setup -DWasPassword=dmgr_password task from the wp_profile_root/ConfigEngine directory to create the dynamic cluster.

   Note: This task may display a warning message since the cluster already exists but it will proceed to create the new dynamic cluster; therefore, the warning can be ignored.

8. Perform the following steps to access the Web Content Management content through an external Web server:

A. Log on to the deployment manager administrative console.

B. Select Environment > WebSphere Variables.

C. From the Scope drop-down menu, select the Node=nodename, Server=servername option to narrow the scope of the listed variables, where Node=nodename is the node that contains the application server.

D. Update the WCM_HOST variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.

E. Update the WCM_PORT variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.

F. Perform the following steps to synchronize the node with the deployment manager:

   1. Select System Administration > Nodes.
   2. Select the node that you want to synchronize from the list.
   3. Click Full Resynchronize.

G. Log off of the deployment manager administrative console.

9. Run the following tasks to propagate your changes:

   Note: WebSphere_Portal_nodename is the name of the node's WebSphere Portal server; but if you customized the server name, the default name changes. If you are uncertain of the
server name, run the `serverstatus -all` task to get a list of the server names and their status.

A. `./stopManager.sh -username admin_userid -password admin_password`, from the `dmgr_profile_root/bin` directory

B. `./stopNode.sh -username admin_userid -password admin_password`, from the `wp_profile_root/bin` directory

C. `./stopServer.sh WebSphere_Portal_nodename -username admin_userid -password admin_password`

D. `./startManager.sh`, from the `dmgr_profile_root/bin` directory

E. `./startNode.sh`, from the `wp_profile_root/bin` directory

F. `./startServer.sh WebSphere_Portal_nodename`

**Parent topic:** Choosing the type of cluster to create on HP-UX
Preparing a remote Web server when portal is installed on HP-UX

Install and configure the Web server plug-in, provided by IBM® WebSphere® Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

- **Prerequisites**
  - Preparing the WebSphere Application Server Deployment Manager on HP-UX
  - Preparing prerequisite and corequisite software on HP-UX
  - Preparing your HP-UX operating system
  - Preparing the primary node on HP-UX
  - Choosing the type of cluster to create on HP-UX

Perform the following steps to install and configure your Web server:

1. Install and configure the Web server; refer to the Web server documentation for information.
2. If using IBM Lotus® Domino®, edit the `NOTES.INI` file on the Web server. Set the `HTTPEnableConnectorHeaders` parameter to 1.
3. If using IBM HTTP Server or Apache Server, edit the `httpd.conf` file on the Web server. Set the `AllowEncodedSlashes` directive to `on`; the directive should be added at the root level as a global directive.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>See the appropriate HTTP Server documentation</td>
<td>IBM HTTP Server</td>
</tr>
<tr>
<td>See the appropriate Apache Server documentation</td>
<td>AllowEncodedSlashes directives</td>
</tr>
</tbody>
</table>

4. Stop the Web server.
5. Install and configure the Web server plug-in on the system where the Web server is located using the Plug-ins installation wizard provided with WebSphere Application Server. Refer to Selecting a Web server topology diagram and roadmap for information. **Note:** The WebSphere Application Server Web server instructions are for the default profile. If not using the default profile, refer to Configuring a Web server for a non-default profile.

- **If using WebDAV:** After successfully installing the Web server plug-in, locate and open your `plugin-cfg.xml` file and set `AcceptAllContent` to `true`.

  **Important:** Depending on how you use the Web server, you may need to adjust the `ServerIOTimeout` value, which defines how long the plug-in should wait for a response from the application. The recommended minimum value is 60 but you may need to adjust this value higher if you are retrieving data from a database. To update this value, locate and open your `plugin-cfg.xml` file and set `ServerIOTimeout` to a value that is appropriate for your business needs. For additional information, see Common questions about the Web server plug-in.

6. If you are using Sun Web Server Version 7 update 8, read Technote 1448262 and perform the steps to resolve the HTTP 408/409 error.
7. Start the Web server.

**Parent topic:** Preparing prerequisite and corequisite software on HP-UX

**Previous topic:** Preparing the WebSphere Application Server Deployment Manager on HP-UX

**Parent topic:** Setting up a cluster on HP-UX

**Previous topic:** Choosing the type of cluster to create on HP-UX
Next topic: Configuring WebSphere Portal to use a user registry on HP-UX in a clustered environment
Configuring WebSphere Portal to use a user registry on HP-UX in a clustered environment

Configure user registry security on IBM® WebSphere® Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

Prior to configuring security, you should use the IBM WebSphere Application Server backupConfig task to create and store a backup of the IBM WebSphere Portal configuration; see backupConfig command for information.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on HP-UX
  - Preparing your HP-UX operating system
  - Preparing the primary node on HP-UX
  - Choosing the type of cluster to create on HP-UX

Perform the following tasks to configure WebSphere Portal to use a user registry:

- **Preparing user registries on HP-UX**
  Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

- **Choosing your user registry model on HP-UX in a clustered environment**
  Choose between securing IBM WebSphere Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

Parent topic: Setting up a cluster on HP-UX

Previous topic: Preparing a remote Web server when portal is installed on HP-UX

Next topic: Preparing additional nodes on HP-UX
Preparing user registries on HP-UX

Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

Choose the appropriate LDAP server to install and setup:

- **Preparing a Domino Directory server**
  If you plan to use a Domino Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a SecureWay Security Server**
  If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Novell eDirectory**
  If you plan to use a Novell eDirectory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Sun Java System Directory Server**
  If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Tivoli Directory Server**
  If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

**Parent topic:** Configuring WebSphere Portal to use a user registry on HP-UX in a clustered environment
Preparing a Domino Directory server

If you plan to use a Domino Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Domino Directory:

1. Perform the following steps to install the Domino Directory:
   A. Go to Lotus Domino documentation.
   B. Select the appropriate version tab for your product.
   C. Click the Download/View online link for the Lotus Domino for multiple platforms Information Center.
   D. Click Domino Administrator Help > Installation > Installing and setting up Domino servers > Server installation > Installing Domino on Unix and perform this task.
   E. Click Domino Administrator Help > Installation > Installing and setting up Domino servers > The Domino server setup program and perform this task.

2. Perform the following steps as a guide to create the WebSphere Portal administrative user:
   A. Navigate to the People view of the Domino Directory and then click Add Person.
   B. Enter the following values in the New Person form to create the wpsbind user:
      - Last Name
        - wpsbind
      - User Name
        - wpsbind/DominoDomain, where DominoDomain is your Lotus Domino Internet domain
      - wpsbind
        Note: Make sure you enter two values in the User Name field, where the first value includes the Lotus Domino domain.
      - Short name/UserID
        - wpsbind
      - Internet password
        - wpsbind

   C. Click Save and Close to save the new person record for wpsbind and return to the People view.
   D. Click Add Person and enter the following values in the New Person form to create the wpsadmin user:
      - Last Name
        - wpsadmin, where wpsadmin in the user ID for the WebSphere Portal Administrator
      - User Name
        - wpsadmin/DominoDomain, where DominoDomain is your Lotus Domino Internet domain
      - wpsadmin
        Note: Make sure you enter two values in the User Name field, where the first value includes the Lotus Domino domain.
      - Short name/UserID
        - wpsadmin
      - Internet password
        - wpsadmin

   E. Click Save and Close to save the new person record for wpsadmin and return to the People view.
   F. Navigate to the Groups view and click Add Group.
   G. Enter the following values in the New Group form on the Basic tab.
      - Group name
- **wpsadmins**
  
  **Note:** If you plan to configure WebSphere Portal for multiple user registries and your Lotus Domino LDAP will share a realm with another user registry, you must use the hierarchical naming convention for the group names, for example: wpsadmins/DominoDomain, to avoid unexpected results during WebSphere Portal runtime.

  **- Group type**
  
  - Multi-purpose

  **- Members**
  
  - wpsbind/DominoDomain
  - wpsadmin/DominoDomain
  
  **Note:** You can add additional administrator users if required.

**H.** Click **Save and Close** to save the wpsadmins group with the wpsbind and wpsadmin users as members.

3. Perform the following steps to update the access control list for the Domino Directory:

   **A.** Open the `names.nsf` file in the Lotus Domino Administrator or Lotus Notes client.

   **B.** Click **File > Application > Access Control** from the main menu to open the access control list for the file.

   **C.** In the **Access Control List > Basics** panel, ensure that the wpsadmins group has either Author or Editor access.

   **D.** Add the following Role Types to the wpsadmins group:

   - GroupCreator
   - GroupModifier
   - UserCreator
   - UserModifier

   **E.** Click **OK**.

**Parent topic:** Preparing user registries on HP-UX
Preparing a SecureWay Security Server

If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Sun Java System Directory Server:

1. Install Sun Java System Directory Server; refer to Oracle software for information.

2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every dc=yourco,dc=com with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.
   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on HP-UX
Preparing a Novell eDirectory

If you plan to use a Novell eDirectory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Novell eDirectory:

1. Install Novell eDirectory; refer to eDirectory for information.

2. Perform the following steps to create the WebSphere Portal administrative user:

   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.

   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.

   C. Replace every dc=yourco,dc=com with your suffix.

   D. Replace any prefixes and suffixes that are unique to your LDAP server.

   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.

   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupName.

   G. Save your changes.

   H. Follow the instructions provided with your directory server to import the LDIF file.

   I. Stop and restart the LDAP server.
Preparing a Sun Java System Directory Server

If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Sun Java System Directory Server:

1. Install Sun Java System Directory Server; refer to Oracle software for information.

2. Perform the following steps to create the WebSphere Portal administrative user:

   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.

   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.

   C. Replace every dc=yourco,dc=com with your suffix.

   D. Replace any prefixes and suffixes that are unique to your LDAP server.

   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.

   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.

   G. Save your changes.

   H. Follow the instructions provided with your directory server to import the LDIF file.

   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on HP-UX
Preparing a Tivoli Directory Server

If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Tivoli Directory Server:


   **Note:** Due to a restriction in Tivoli Directory Server, users or groups must not contain a Turkish uppercase dotted I or lowercase dotted i in the DN as this will prevent correct retrieval of that user or group.

2. Perform the following steps to create the WebSphere Portal administrative user:

   A. Optional: Perform the following steps to create a new directory suffix:

      1. Click the **Server Administration** folder, located in the left-hand navigation of the directory server console.
      2. Click the **Manage Server Properties** folder under the Server Administration folder and then select **Suffixes** on the main page.
      3. Type the **Base DN** name for the suffix; for example: `dc=yourcompany,dc=com`.
      4. Click **Add**.
      5. Click **OK** to save your changes.
      6. Stop and restart the LDAP server.

   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:

      - Use the **PortalUsers.ldap** file as a working example and adapted appropriately to work with your LDAP server.
      - Use the **ContentUsers.ldap** file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.

   C. Replace every `dc=yourco,dc=com` with your suffix.

   D. Replace any prefixes and suffixes that are unique to your LDAP server.

   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.

   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the **objectclasses** to **accessGroup**. If using Tivoli Access Manager Version 6, set the **objectclasses** to **groupOfNames**.

   G. Save your changes.

   H. Follow the instructions provided with your directory server to import the LDIF file.

   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on HP-UX
Choosing your user registry model on HP-UX in a clustered environment

Choose between securing IBM® WebSphere® Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

Choose one of the following user registry models:

- **Configuring a stand-alone LDAP user registry on HP-UX in a clustered environment**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Configuring the default federated repository on HP-UX in a clustered environment**
  Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

- **Adapting the attribute configuration**
  After installing IBM WebSphere Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

**Parent topic:** Configuring WebSphere Portal to use a user registry on HP-UX in a clustered environment
Configuring a stand-alone LDAP user registry on HP-UX in a clustered environment

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Configuring a stand-alone LDAP user registry on HP-UX in a clustered environment**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry to store all user account information for authorization.

- **Configuring a stand-alone LDAP user registry over SSL on HP-UX in a clustered environment**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

**Parent topic:** Choosing your user registry model on HP-UX in a clustered environment
Configuring a stand-alone LDAP user registry on HP-UX in a clustered environment

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry to store all user account information for authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps. If you need to rerun the wp-modify-ldap-security task to change the LDAP repositories or because the task failed, you must choose a new name for the realm using the standalone.ldap.realm parameter or you can set ignoreDuplicateIds=true in the wkplc.properties file, before rerunning the task.

Perform the following steps to configure a standalone LDAP user registry:

Note: Use the wp_security_xxx.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_security_xxx.properties helper file.

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

2. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM Stand-alone LDAP configuration heading:

- standalone.ldap.id
- standalone.ldap.host
- standalone.ldap.port
- standalone.ldap.bindDN
- standalone.ldap.bindPassword
- standalone.ldap.ldapServerType
- standalone.ldap.userIdMap
- standalone.ldap.groupIdMap
- standalone.ldap.groupMemberIdMap
- standalone.ldap.userFilter
- standalone.ldap.groupFilter
- standalone.ldap.serverId
- standalone.ldap.serverPassword
- standalone.ldap.realm
- standalone.ldap.primaryAdminId
- standalone.ldap.primaryAdminPassword
- standalone.ldap.primaryPortalAdminId
- standalone.ldap.primaryPortalAdminPassword
- standalone.ldap.primaryPortalAdminGroup
- standalone.ldap.baseDN
3. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.idap.et.group.objectClasses`
   - `standalone.idap.et.group.objectClassesForCreate`
   - `standalone.idap.et.group.searchBases`
   - `standalone.idap.et.personaccount.objectClasses`
   - `standalone.idap.et.personaccount.objectClassesForCreate`
   - `standalone.idap.et.personaccount.searchBases`

4. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attributes heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.idap.gm.groupMemberName`
   - `standalone.idap.gm.objectClass`
   - `standalone.idap.gm.scope`
   - `standalone.idap.gm.dummyMember`

5. Required: Enter a value for the following required relative distinguished name (RDN®) parameters in the `wkplc.properties` file under the Default parent, RDN attribute heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.idap.personAccountParent`
   - `standalone.idap.groupParent`
   - `standalone.idap.personAccountRdnProperties`
   - `standalone.idap.groupRdnProperties`

6. Save your changes to the `wkplc.properties` file.

7. Optional: Enter the following Web content authors parameters in the `wkplc_comp.properties` file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. **Note:** See the `wkplc_comp.properties` file for specific information about the required parameters and for advanced parameters.
   - `WcmContentAuthorsGroupId`
   - `WcmContentAuthorsGroupCN`

8. Run the `./ConfigEngine.sh validate-standalone-ldap -DWasPassword=password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, `WasPassword` is the value entered during installation and not a value found in your LDAP user registry. **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now? Press y then Enter.

9. Run the `./ConfigEngine.sh wp-modify-ldap-security -DWasPassword=password` task, from the `wp_profile_root` /ConfigEngine directory, to set the stand-alone LDAP user registry.

10. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

11. Run the `./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root` /ConfigEngine directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

12. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the
LDAP directory.

**Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

A. Edit the `wp_profile_root`/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties file.

B. Add the following lines to the file:

```
uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
```

Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `./ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.idp.realm</code> in the wkplc.properties file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the wkplc.properties file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

13. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring a stand-alone LDAP user registry on HP-UX in a clustered environment

**Related tasks**

Starting and stopping servers, deployment managers, and node agents

Enabling LDAP security after cluster creation
Configuring a stand-alone LDAP user registry over SSL on HP-UX in a clustered environment

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to configure a standalone LDAP user registry over SSL: **Note:** Use the **wp_security_xxx.properties** helper file, located in the **wp_profile_root/ConfigEngine/config/helpers** directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the **wkplc.properties** file, you will use your **wp_security_xxx.properties** helper file.

1. To specify the LDAP server’s SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:
   - **Server trust store**
     A. Add the certificate to the trust store:
        1. Log in to the WebSphere Application Server Administrative Console.
        2. Navigate to **Security > SSL certificate and key management > SSL configurations**.
        3. Click the appropriate SSL configuration from the list. For example,
           - Stand-alone environments: **NodeDefaultSSLSettings**
           - Clustered environments: **CellDefaultSSLSettings**
        4. **Clustered environments:** Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.
        5. Click **Key stores and certificates**.
        6. Click **Signer certificates**, click **Add**, and then enter the following information:
           - Type the **Alias** the key store uses for the signer certificate.
           - Type the **File name** where the signer certificate is located.
        7. Click **OK** and then click **Save** to save the changes to the master configuration.
   B. Retrieve the certificate from the port:
      1. Log in to the WebSphere Application Server Administrative Console.
      2. Navigate to **Security > SSL certificate and key management > SSL configurations**.
      3. Click the appropriate SSL configuration from the list. For example,
         - Stand-alone environments: **NodeDefaultSSLSettings**
- Clustered environments: **CellDefaultSSLSettings**

**Clustered environments:** Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.

4. Click **Key stores and certificates**.

5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.

6. Click **Signer certificates**, click **Retrieve from port**, and then enter the following information:
   - Type the **Host** name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the **Port** used when attempting to retrieve the signer certificate.
   - Type the **Alias** the key store uses for the signer certificate.

7. Click **Retrieve signer information** to retrieve the certificate from the port.

8. Click **OK** and then click **Save** to save the changes to the master configuration.

- **Client trust store**
  - **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

A. See **Secure installation for client signer retrieval**.

B. Run the **retrieveSigners** task from the **wp_profile_root/bin** directory; see **retrieveSigners command** for information. In a deployed environment, you will need to run the **retrieveSigners** task, for any federated node, against the Deployment Manager. **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

C. Update the trust store properties file:

1. Use a text editor to open the **ssl.client.props** file, located in the **wp_profile_root/properties** directory.

2. Change the **com.ibm.ssl.trustStore** parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter **com.ibm.ssl.trustStore=${CONFIG_ROOT}\cells\wpsbvt\nodes\wpsbvt\trust.p12** to use the default trust store.

3. Save your changes.

2. Use a text editor to open the **wkplc.properties** file, located in the **wp_profile_root/ConfigEngine/properties** directory.

3. Required: Enter a value for the following required parameters in the **wkplc.properties** file under the VMM Standalone LDAP configuration heading. **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - **standalone.idap.id**
   - **standalone.idap.host**
   - **standalone.idap.port**
   - **standalone.idap.blndDN**
   - **standalone.idap.bindPassword**
   - **standalone.idap.idapServerType**
   - **standalone.idap.useridMap**
   - **standalone.idap.groupldMap**
   - **standalone.idap.groupMemberldMap**
   - **standalone.idap.userFilter**
   - **standalone.idap.groupFilter**
   - **standalone.idap.serverld**
   - **standalone.idap.serverPassword**
   - **standalone.idap.realm**
   - **standalone.idap.primaryAdminId**
   - **standalone.idap.primaryAdminPassword**
   - **standalone.idap.primaryPortalAdminId**
4. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading:\textbf{Note:} See the properties file for specific information about the required parameters and for advanced parameters.

- `standalone.ldap.et.group.objectClasses`
- `standalone.ldap.et.group.objectClassesForCreate`
- `standalone.ldap.et.group.searchBases`
- `standalone.ldap.et.personaccount.objectClasses`
- `standalone.ldap.et.personaccount.objectClassesForCreate`
- `standalone.ldap.et.personaccount.searchBases`

5. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attributes heading: \textbf{Note:} See the properties file for specific information about the required parameters and for advanced parameters.

- `standalone.ldap.gm.groupMemberName`
- `standalone.ldap.gm.objectClass`
- `standalone.ldap.gm.scope`
- `standalone.ldap.gm.dummyMember`

6. Required: Enter a value for the following required relative distinguished name (RDN®) parameters in the `wkplc.properties` file under the Default parent, RDN attribute heading: \textbf{Note:} See the properties file for specific information about the required parameters and for advanced parameters.

- `standalone.ldap.personAccountParent`
- `standalone.ldap.groupParent`
- `standalone.ldap.personAccountRdnProperties`
- `standalone.ldap.groupRdnProperties`

7. Enter a value for the following parameters to enable Secure Socket Layers (SSL): \textbf{Note:} See the properties file for specific information about the required parameters and for advanced parameters.

Required parameters:

- `standalone.ldap.sslEnabled`
- `standalone.ldap.sslConfiguration`

Optional parameters:

- `standalone.ldap.certificateMapMode`
- `standalone.ldap.certificateFilter`

8. Save your changes to the `wkplc.properties` file.

9. Run the `./ConfigEngine.sh validate-standalone-ldap -DWasPassword=password` task to validate your LDAP server settings. \textbf{Attention:} If you have not deleted the default file repository, \textbf{WasPassword} is the value entered during installation and not a value found in your LDAP user registry.

\textbf{Note:} During the validation task, you may receive the following prompt: \textit{Add signer to the trust store now?}. Press `y` then \textbf{Enter}.

10. Run the `./ConfigEngine.sh wp-modify-ldap-security -DWasPassword=password` task, from the \textit{wp_profile_root}/ConfigEngine directory, to set the stand-alone LDAP user registry.

11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: \textit{Starting and stopping servers, deployment managers, and node agents}.

12. Run the `./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password` task, from the \textit{wp_profile_root}/ConfigEngine directory, to check that all defined attributes are available in the configured
LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring a stand-alone LDAP user registry on HP-UX in a clustered environment

**Related tasks**

- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation
Configuring the default federated repository on HP-UX in a clustered environment

Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

Perform the following tasks to add user registries to the default federated repository:

- **Configuring a federated LDAP user registry on HP-UX in a clustered environment**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Adding a database user registry on HP-UX in a clustered environment**
  Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

- **Adding realm support on HP-UX in a clustered environment**
  A realm is a group of users from one or more user registries that form a coherent group within IBM WebSphere Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

Parent topic: Choosing your user registry model on HP-UX in a clustered environment
Configuring a federated LDAP user registry on HP-UX in a clustered environment

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Adding an LDAP user registry on HP-UX in a clustered environment**
  
  Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time. If IBM Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=groupName` and the hierarchical format is `cn=groupName,o=root`. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as `wpsadmin`, this ID cannot exist in the LDAP you are adding.

- **Adding an LDAP user registry over SSL on HP-UX in a clustered environment**
  
  Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

**Parent topic:** Configuring the default federated repository on HP-UX in a clustered environment
Adding an LDAP user registry on HP-UX in a clustered environment

Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time. If IBM® Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=groupName` and the hierarchical format is `cn=groupName,o=root`. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as wpsadmin, this ID cannot exist in the LDAP you are adding.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

**Tip:** Perform these steps on the primary node only.

**Note:** Use the `wp_add_federated_xxx.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_add_federated_xxx.properties` helper file.

1. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
2. Required: Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated LDAP Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `federated.ldap.id`
   - `federated.ldap.host`
   - `federated.ldap.port`
   - `federated.ldap.bindDN`
   - `federated.ldap.bindPassword`
   - `federated.ldap.ldapServerType`
   - `federated.ldap.baseDN`
3. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `federated.ldap.et.group.objectClasses`
   - `federated.ldap.et.group.objectClassesForCreate`
   - `federated.ldap.et.group.searchBases`
   - `federated.ldap.et.personaccount.objectClasses`
4. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attribute heading: 

```
- federated.ldap.et.personaccount.objectClassesForCreate
- federated.ldap.et.personaccount.searchBases
```

Note: See the properties file for specific information about the required parameters and for advanced parameters.

```
- federated.ldap.gm.groupMemberName
- federated.ldap.gm.objectClass
- federated.ldap.gm.scope
- federated.ldap.gm.dummyMember
```

5. Save your changes to the `wkplc.properties` file.

6. Optional: Enter the following Web content authors parameters in the `wkplc_comp.properties` file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. 

Note: See the `wkplc_comp.properties` file for specific information about the required parameters and for advanced parameters.

```
- WcmContentAuthorsGroupId
- WcmContentAuthorsGroupCN
```

7. Run the `./ConfigEngine.sh validate-federated-ldap -DWasPassword=password` task to validate your LDAP server settings. 

Attention: If you have not deleted the default file repository, `WasPassword` is the value entered during installation and not a value found in your LDAP user registry.

Note: During the validation task, you may receive the following prompt: Add signer to the trust store now? . Press y then Enter.

8. Run the `./ConfigEngine.sh wp-create-ldap -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to add an LDAP user registry to the default federated repository. 

Note: Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

10. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms: 

Note: See the properties file for specific information about the required parameters and for advanced parameters.

```
- id
- baseDN
- nameInRepository
```

C. Save your changes to the `wkplc.properties` file.

D. Run the `./ConfigEngine.sh wp-create-base-entry -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to create a base entry in a repository.

E. Stop and restart all necessary servers to propagate your changes.

11. Optional: Run the `./ConfigEngine.sh wp-query-repository -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to list the names and types of configured repositories.

12. Run the `./ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured
LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

13. Perform the following steps to update the user registry where new users and groups are stored:

   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - `personAccountParent`
      - `groupParent`
      - `personAccountRdnProperties`
      - `groupRdnProperties`

      The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:

      - `personAccountParent=dc=yourco,dc=com`
      - `groupParent=dc=yourco,dc=com`

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to delete the old attributes before adding the new attributes.

   E. Stop and restart all necessary servers to propagate your changes.

14. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for `realmName` or leave blank to update the default realm.

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

   E. Stop and restart all necessary servers to propagate your changes.

15. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

   **Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

   A. Edit the `wp_profile_root/PortalServer/wcm/shared/app/config/wcm/services/MemberFixerModule.properties` file.

   B. Add the following lines to the file:

   ```
   uid=xyzadmin, o=defaultWIMFileBasedRealm -> portal_admin_DN
   cn=contentauthors, o=defaultWIMFileBasedRealm -> content_authors_group_DN
   ```
Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `./ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management.

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

16. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

17. Optional: **This step is required in a production environment.** Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

   **Important:**
   - Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
   - Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

   **Note:** If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

   A. Run the following task from the `wp_profile_root/ConfigEngine` directory: `./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword`. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` parameter.

   B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

   C. Run the following task from the `wp_profile_root/ConfigEngine` directory: `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid`. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

   D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.
18. Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it. The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: *Deleting the repository.*

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring a federated LDAP user registry on HP-UX in a clustered environment

**Related tasks**

- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation

**Related information**

- User IDs and passwords
Adding an LDAP user registry over SSL on HP-UX in a clustered environment

Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry over SSL to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

Tip: Perform these steps on the primary node only.

Note: Use the wp_add_federated_xxx.properties helper file, located in the wp_profile_root /ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_add_federated_xxx.properties helper file.

1. To specify the LDAP server's SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:
   - **Server trust store**
     A. Add the certificate to the trust store:
        1. Log in to the WebSphere Application Server Administrative Console.
        2. Navigate to Security > SSL certificate and key management > SSL configurations.
        3. Click the appropriate SSL configuration from the list. For example,
           - Stand-alone environments: NodeDefaultSSLSettings
           - Clustered environments: CellDefaultSSLSettings

        **Clustered environments:** Ensure the setting for SSL configuration for outbound connection matches your SSL settings.
        4. Click Key stores and certificates.
        5. Click the appropriate trust store from the list; for example, NodeDefaultTrustStore.
        6. Click Signer certificates, click Add, and then enter the following information:
           - Type the Alias the key store uses for the signer certificate.
           - Type the File name where the signer certificate is located.
        7. Click OK and then click Save to save the changes to the master configuration.
     B. Retrieve the certificate from the port:
        1. Log in to the WebSphere Application Server Administrative Console.
2. Navigate to **Security > SSL certificate and key management > SSL configurations**.

3. Click the appropriate SSL configuration from the list. For example,
   - Stand-alone environments: **NodeDefaultSSLSettings**
   - Clustered environments: **CellDefaultSSLSettings**

   **Clustered environments:** Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.

4. Click **Key stores and certificates**.

5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.

6. Click **Signer certificates**, click **Retrieve from port**, and then enter the following information:
   - Type the **Host** name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the SSL **Port** used when attempting to retrieve the signer certificate.
   - Type the **Alias** the key store uses for the signer certificate.

7. Click **Retrieve signer information** to retrieve the certificate from the port.

8. Click **OK** and then click **Save** to save the changes to the master configuration.

   **Client trust storeNote:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

   A. See **Secure installation for client signer retrieval**.

   B. Run the **retrieveSigners** task from the **wp_profile_root/bin** directory; see **retrieveSigners command** for information. In a deployed environment, you will need to run the retrieveSigners task, for any federated node, against the Deployment Manager. **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

C. Update the trust store properties file:

   1. Use a text editor to open the **ssl.client.props** file, located in the **wp_profile_root/properties** directory.
   2. Change the **com.ibm.ssl.trustStore** parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter

   ```
   com.ibm.ssl.trustStore=${CONFIG_ROOT}\cells\wpsbvt\nodes\wpsbvt\trust.p12
   ```

   to use the default trust store.
   3. Save your changes.

2. Use a text editor to open the **wkplc.properties** file, located in the **wp_profile_root/ConfigEngine/properties** directory.

3. Required: Enter a value for the following required parameters in the **wkplc.properties** file under the VMM Federated LDAP Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - **federated.idap.id**
   - **federated.idap.host**
   - **federated.idap.port**
   - **federated.idap.blndDN**
   - **federated.idap.blndPassword**
   - **federated.idap.idapServerType**
   - **federated.idap.baseDN**

4. Required: Enter a value for the following required entity types parameters in the **wkplc.properties** file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - **federated.idap.et.group.objectClasses**
   - **federated.idap.et.group.objectClassesForCreate**
   - **federated.idap.et.group.searchBases**
5. Required: Enter a value for the following required group member parameters in the wkplc.properties file under the Group member attribute heading: 

- federated.ldap.gm.groupMemberName
- federated.ldap.gm.objectClass
- federated.ldap.gm.scope
- federated.ldap.gm.dummyMember

Note: See the properties file for specific information about the required parameters and for advanced parameters.

6. Enter a value for the following parameters to enable Secure Socket Layers (SSL):

- federated.ldap.sslEnabled
- federated.ldap.sslConfiguration

Optional parameters:

- federated.ldap.certificateMapMode
- federated.ldap.certificateFilter

7. Save your changes to the wkplc.properties file.

8. Optional: Enter the following Web content authors parameters in the wkplc_comp.properties file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. 

- WcmContentAuthorsGroupId
- WcmContentAuthorsGroupCN

Note: See the wkplc_comp.properties file for specific information about the required parameters and for advanced parameters.

9. Run the .//ConfigEngine.sh validate-federated-ldap -DWasPassword=Password task to validate your LDAP server settings. Attention: If you have not deleted the default file repository, WasPassword is the value entered during installation and not a value found in your LDAP user registry.

Note: During the validation task, you may receive the following prompt: Add signer to the trust store now? Press y then Enter.

10. Run the .//ConfigEngine.sh wp-create-ldap -DWasPassword=Password task, from the wp_profile_root/ConfigEngine directory, to add an LDAP user registry to the default federated repository. Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

12. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:

A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

B. Enter a value for the following required parameters in the wkplc.properties file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms.

Note: See the properties file for specific information about the required parameters and for advanced parameters.
- Id
- baseDN
- nameInRepository

C. Save your changes to the wkplc.properties file.

D. Run the `./ConfigEngine.sh wp-create-base-entry -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to create a base entry in a repository.

E. Stop and restart all necessary servers to propagate your changes.

13. Optional: Run the `./ConfigEngine.sh wp-query-repository -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to list the names and types of configured repositories.

14. Run the `./ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

15. Perform the following steps to update the user registry where new users and groups are stored:

   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:
      - `personAccountParent`
      - `groupParent`
      - `personAccountRdnProperties`
      - `groupRdnProperties`

      The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:
      - `personAccountParent=dc=yourco,dc=com`
      - `groupParent=dc=yourco,dc=com`

      C. Save your changes to the `wkplc.properties` file.

      D. Run the `./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to delete the old attributes before adding the new attributes.

      E. Stop and restart all necessary servers to propagate your changes.

16. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for `realmName` or leave blank to update the default realm.

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.
E. Stop and restart all necessary servers to propagate your changes.

17. Configure your Web server over SSL. Navigate to Configuring WebSphere Portal > Additional security features > Configuring SSL > Setting up SSL for information.

18. Log on to the WebSphere Application Server Administrative Console and navigate to Security > SSL certificate and key management. Click the Use the United States Federal Information Processing Standard (FIPS) algorithms. check box to enable FIPS.

19. Enable TLS in your internet browser, located under Tools > Options > Advanced.

20. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

   Note: This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the configure-express task.

   A. Edit the wp_profile_root
   /PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties file.

   B. Add the following lines to the file:

   ```
   uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
   cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
   ```

   Replace portal_admin_DN with the distinguished name of the portal administrator and content_authors_group_DN with the distinguished name of the content authors group used during LDAP configuration.

   C. Save your changes and close the file.

   D. Run the "./ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password" task, located in the wp_profile_root/ConfigEngine directory. Note: Choose the appropriate value to enter for realm_name depending on the type of LDAP user registry you configured: Table 1. Value for realm_name when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for realm_name should match the value for standalone.ldap.realm in the wkplc.properties file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for realm_name should match the value for federated.realm in the wkplc.properties file. If the value for federated.realm is empty, use defaultWIMFileBasedRealm as the default value.</td>
</tr>
</tbody>
</table>

21. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

22. Optional: This step is required in a production environment. Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

   Important:
   - Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
   - Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.
**Note:** If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

A. Run the following task from the `wp_profile_root/ConfigEngine` directory:

```
./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword
```

**Important:** You must provide the full distinguished name (DN) for the `newAdminId` parameter.

B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

C. Run the following task from the `wp_profile_root/ConfigEngine` directory:

```
./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid
```

**Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

23. Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it. The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: *Deleting the repository.*

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring a federated LDAP user registry on HP-UX in a clustered environment

**Related tasks**

- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation

**Related Information**

- User IDs and passwords
Adding a database user registry on HP-UX in a clustered environment

Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps. If you have WebSphere Application Server Version 7.0.x installed, you must install APAR PM23090 and APAR PM24181 for WebSphere Portal prior to running this task. Clusters with WebSphere Application Server Version 6.1.x do not require these APARs.

Perform the following steps to add a database user registry to the default federated repository; you must repeat these steps for each additional database user registry that you plan to add:

**Tip:** Perform these steps on the primary node only.

**Note:** Use the `wp_add_DB.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_add_DB.properties` helper file.

1. Set up a new database, including creating a new user with appropriate database privileges for accessing the database:

   **Instructions for setting up databases:** Refer to the appropriate documentation for the type of database you want to set up.

   **Consulting your database administrator:** The task of setting up a new database is typically performed by a database administrator. However, the following steps are provided for your reference in the event you are creating a stand-alone database for testing or demonstration purposes. Consult your database administrator before proceeding with the following steps if you plan to create a database for a production environment.

   **Table 1. Steps for creating a new database to use as a database user registry.**

<table>
<thead>
<tr>
<th>Database</th>
<th>Steps</th>
</tr>
</thead>
</table>

1441
<table>
<thead>
<tr>
<th>Database Type</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DB2</strong></td>
<td>Perform the following steps to create a DB2 database: Install DB2. Enter the following database tuning commands: db2 &quot;CREATE DB dbname using codeset UTF-8 territory us PAGESIZE 8192&quot; db2 &quot;UPDATE DB CFG FOR dbname USING appheapsz 4096&quot; db2 &quot;UPDATE DB CFG FOR dbname USING app_ctl_heap_sz 1024&quot; db2 &quot;UPDATE DB CFG FOR dbname USING stmtheap size 32768&quot; db2 &quot;UPDATE DB CFG FOR dbname USING dbheap 1400&quot; db2 &quot;UPDATE DB CFG FOR dbname USING locklist 1000&quot; db2 &quot;UPDATE DB CFG FOR dbname USING logfilsiz 4000&quot; db2 &quot;UPDATE DB CFG FOR dbname USING logprimary 12&quot; db2 &quot;UPDATE DB CFG FOR dbname USING logsecond 20&quot; db2 &quot;UPDATE DB CFG FOR dbname USING avg_appls 5&quot; db2 &quot;UPDATE DB CFG FOR dbname USING locktimeout 30&quot; db2 &quot;UPDATE DB CFG FOR dbname USING AUTO_MAINT off&quot;</td>
</tr>
<tr>
<td><strong>Oracle</strong></td>
<td>Perform the following steps to create an Oracle database: Install Oracle using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16. Configure the database in Dedicated Server Mode. Enter the recommended initial buffer pool sizes or set them according to your business needs: <code>db_block_size = 8192</code>, <code>db_cache_size = 300M</code>, <code>db_files = 1024</code>, <code>log_buffer = 65536</code>, <code>open_cursors = 1500</code>, <code>pga_aggregate_target = 200M</code>, <code>pre_page_sga = true</code>, <code>processes = 300</code>, <code>shared_pool_size = 200M</code></td>
</tr>
<tr>
<td><strong>SQL Server</strong></td>
<td>Perform the following steps to create an SQL Server database: Install SQL Server. Set <strong>Collation</strong> to <strong>case-sensitive</strong>. <strong>Note:</strong> Install SQL Server with the appropriate portal database collation so that your tempdb collation setting matches the collation you use for the property extension database. The tempdb collation is inherited from the master database, which you set when you install SQL Server.</td>
</tr>
</tbody>
</table>

2. Perform the following steps to define the **DbDriver** and **DbLibrary** parameter values:
   A. Navigate to the following directory: `wp_profile_root/ConfigEngine/properties`  
   B. Locate and open `wkplc_dbtype.properties` with any text editor.  
   C. Enter a value for the following parameters under the appropriate database type properties heading:  
      - `db_type.DbDriver`  
      - `db_type.DbLibrary`  
   D. Save your changes.

3. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

4. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated Database Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
5. Change the value for the \texttt{com.ibm.SOAP.requestTimeout} parameter to 1000.
   A. Navigate to the following directory: \texttt{wp_profile_root/properties}
   B. Locate and open \texttt{soap.client.props} with any text editor.
   C. Locate the \texttt{com.ibm.SOAP.requestTimeout} parameter and change the value to 1000.
   D. Save and close \texttt{soap.client.props}.

6. Perform the following steps in a clustered environment:
   A. Run the \texttt{/ConfigEngine.sh wp-prep-vmm-db-secured-environment -DWasPassword=\textit{password} -DDbDomain=federated.db -D\texttt{db_type}.DmgrDbLibrary=\textit{local path of the database jars on the Deployment Manager} -DDmgrnodeName=\textit{dmgr_node_name}} task from the \texttt{wp_profile_root/ConfigEngine} directory to create the local Deployment Manager WebSphere variable used to access the database jars. \textbf{Note:} The \texttt{db_type} in \texttt{db_type}.\texttt{DmgrDbLibrary} should be set to the type of database you are using, for example \texttt{db2}. The \texttt{local full path of the database jars on the Deployment Manager} should be one of the following options:
      - \texttt{DB2 Type 2 driver:db2java.zip}
      - \texttt{DB2 Type 4 driver:db2jcc.jar;db2jcc_license_cu.jar}
      - \texttt{DB2 for z/OS Type 2 driver:db2java.zip}
      - \texttt{DB2 for z/OS Type 4 driver:db2jcc.jar;db2jcc_license_cisuz.jar;db2jcc javax.jar}
      - \texttt{Oracle:ojdbc14.jar}
      - \texttt{SQL Server JDBC driver provided by Microsoft:sqljdbc.jar}
      - \texttt{SQL Server JDBC driver provided by DataDirect:sqlserver.jar;base.jar;util.jar}
   B. Run the following task. Include each node name as a comma separated list in the command:
      \textbf{Running the task:} You do not have to run this task more than once. You can run this task from any node in the cluster.
      1. Set the property value for \texttt{federated.dbDbType} in the \texttt{wkplc.properties} file if using a database user registry.
      2. Run the \texttt{/ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=\textit{password} -DDbDomain=federated.db -DVmmNodeName=\textit{node_name} -D\texttt{db_type}.NodeDbLibrary=\textit{local full path of the database jars}} task from the \texttt{wp_profile_root/ConfigEngine} directory on each node to create the variable used to access the VMM database jars. \textbf{Note:} The \texttt{VmmNodeName} is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The \texttt{db_type} in \texttt{db_type}.\texttt{NodeDbLibrary} should be set to the type of database you are using, for example \texttt{db2}.
   C. Stop and restart all necessary servers to propagate your changes.

7. Run the \texttt{/ConfigEngine.sh wp-create-db -DWasPassword=\textit{password}} task, from the \texttt{wp_profile_root/ConfigEngine} directory, to add a database user registry to the default federated repository. \textbf{Note:} Users who are not in an LDAP do
8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

9. Perform the following steps to update the user registry where new users and groups are stored:

   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - `personAccountParent`
      - `groupParent`
      - `personAccountRdnProperties`
      - `groupRdnProperties`

      The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:

      - `personAccountParent=dc=yourco,dc=com`
      - `groupParent=dc=yourco,dc=com`

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `. /ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to delete the old attributes before adding the new attributes.

   E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Run the `. /ConfigEngine.sh wp-query-repository -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to list the names and types of configured repositories.

    If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring the default federated repository on HP-UX in a clustered environment

**Related tasks**

Starting and stopping servers, deployment managers, and node agents

Enabling LDAP security after cluster creation
Adding realm support on HP-UX in a clustered environment

A realm is a group of users from one or more user registries that form a coherent group within IBM® WebSphere® Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

Before configuring realm support, you must add all LDAP user registries and/or database user registries, that you will use to create a single realm or multiple realms, to the federated repository. If you are going to create multiple realms, you must create all required base entries within your LDAP user registries and/or database user registries. All base entry names must be unique within the federated repository. In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add realm support to your user registry model:

1. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
2. Required: Enter a value for the following required parameters in the `wkplc.properties` file under the VMM realm configuration heading:
   - `realmName`
   - `securityUse`
   - `delimiter`
   - `addBaseEntry`
   Note: See the properties file for specific information about the required parameters and for advanced parameters.
3. Save your changes to the `wkplc.properties` file.
4. Run the `./ConfigEngine.sh wp-create-realm -DWasPassword=[password] task`, from the `wp_profile_root/ConfigEngine` directory, to add a new realm to the Virtual Member Manager configuration. Important: To create multiple realms, ensure that your federated repository contains the required unique base entries. Stop and restart the appropriate servers for your installation environment, and then update the `wkplc.properties` file with the base entry information and rerun the `wp-create-realm` task. Repeat these steps until all realms are created.
5. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.
6. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM realm configuration heading and then save your changes:
   - `realmName`
   - `realm.personAccountParent`
   - `realm.groupParent`
   - `realm.orgContainerParent`
   Note: See the properties file for specific information about the required parameters and for advanced parameters.
7. Run the `./ConfigEngine.sh wp-modify-realm-defaultparents -DWasPassword=[password]` task, from the `wp_profile_root/ConfigEngine` directory, to update the default parents per entity type and realm. Important: Stop and restart the appropriate servers for your installation environment before rerunning this task for any additional entity types.
and realms.

8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

9. Optional: Perform the following steps to add additional base entries to the realm configuration; for example, if you had two additional base entries (base entry 1 and base entry 2) to add to the realm you just created, you would update the wkplc.properties file with the information from base entry 1 and then run this task. Then you would update the properties file with the information for base entry 2 and then run this task:

A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

B. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading:

   - realmName
   - addBaseEntry

C. Save your changes to the wkplc.properties file.

D. Run the ./ConfigEngine.sh wp-add-realm-baseentry -DWasPassword=password task, from the wp_profile_root /ConfigEngine directory, to add additional LDAP base entries to the realm configuration.

E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID; this step is required if you change the default realm:

A. Create a new user in the Manage Users and Groups portlet to replace the current WebSphere Application Server administrative user.

B. Create a new user in the Manage Users and Groups portlet to replace the current WebSphere Portal administrative user.

C. Create a new group in the Manage Users and Groups portlet to replace the current group.

D. Run the ./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminId -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid task, from the wp_profile_root/ConfigEngine directory, to replace the old WebSphere Application Server administrative user ID and group ID with the new user and group. **Important:** You must provide the full distinguished name (DN) for the newAdminId and newAdminGroupId parameters.

   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the -Dskip.ldap.validation=true parameter to the task to skip the validation.

E. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

F. Run the ./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminId -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid task to replace the old WebSphere Portal administrative user ID and group ID with the new user and group. **Important:** You must provide the full distinguished name (DN) for the newAdminId and newAdminGroupId parameters.

   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the -Dskip.ldap.validation=true parameter to the task to skip the validation.

G. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

11. Optional: Perform the following steps to set the realm you created as the default realm: **Remember:** Only users defined in base entries that exist in the default realm are able to log into WebSphere Portal. If you find that a user cannot log in to WebSphere Portal, check to see if the base entry that contains the user exists in the default realm. You can run the wp-query-realm-baseentry task to see what base entries are part of the default realm. If the default realm is missing
the base entry, run the wp-add-realm-baseentry task to add the base entry to the default realm.

A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

B. For defaultRealmName, type the realmName property value you want to use as the default realm.

C. Save your changes to the wkplc.properties file.

D. Run the./ConfigEngine.sh wp-default-realm -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to set this realm as the default realm.

E. Stop and restart all necessary servers to propagate your changes.

12. Optional: Perform the following steps to query a realm for a list of its base entries:

   A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

   B. For realmName, type the name of the realm you want to query.

   C. Save your changes to the wkplc.properties file.

   D. Run the ./ConfigEngine.sh wp-query-realm-baseentry -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to list the base entries for a specific realm.

13. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

   A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

   B. Enter a value for realmName or leave blank to update the default realm.

   C. Save your changes to the wkplc.properties file.

   D. Run the ./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password task, located in the wp_profile_root/ConfigEngine directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the ./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password task to disable the feature.

   E. Stop and restart all necessary servers to propagate your changes.

If you created your clustered environment then performed the steps in this task, you must now run the enable-jcr-security task on the secondary node. See Enabling LDAP security after cluster creation for instructions.

**Parent topic:** Configuring the default federated repository on HP-UX in a clustered environment

**Related tasks**

Starting and stopping servers, deployment managers, and node agents

Enabling LDAP security after cluster creation

| | | 1447 |
Adapting the attribute configuration

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

After installation, IBM WebSphere Portal has a predefined set of attributes for users and groups. Your LDAP server may have a different set of predefined user and group attributes. To ensure proper communication between WebSphere Portal and your LDAP server, you can configure additional attributes and flag existing attributes as required or unsupported on a per repository basis or for all configure repositories.

LDAP servers can only handle attributes that are explicitly defined in their schema. The LDAP server’s schema is different from the WebSphere Portal schema but the two schemas should match for proper communication between WebSphere Portal and the LDAP server. The task to add the LDAP user registry does some basic attribute configurations depending on the type of LDAP server that you choose. You may, however, still need to adapt the WebSphere Portal configuration to match the LDAP schema; for example, if an attribute is defined in WebSphere Portal but not in the LDAP server, you will need to perform one of the following tasks to resolve this mismatch:

- Flag the attribute as unsupported for the LDAP server
- Introduce an attribute mapping that maps the WebSphere Portal attribute to an attribute defined in the LDAP schema

Perform the following tasks to adapt the attribute configuration to match the configured LDAP server(s) and your business needs:

1. **Querying the defined attributes on HP-UX**
   After installing IBM WebSphere Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

2. **Adding attributes on HP-UX in a clustered environment**
   The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM WebSphere Portal and your user registry.

3. **Mapping attributes on HP-UX in a clustered environment**
   After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

4. **Removing attributes**
   Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

**Parent topic:** Choosing your user registry model on HP-UX in a clustered environment

**Related tasks**

- Adding an LDAP user registry on HP-UX
- Adding an LDAP user registry over SSL on HP-UX
- Configuring a stand-alone LDAP user registry on HP-UX
Configuring a stand-alone LDAP user registry over SSL on HP-UX
Querying the defined attributes on HP-UX

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

**Warning:** If you are using a database user registry or a property extension database, copy the database drivers to the `AppServer_root/lib` directory before executing this script.

Run the `./ConfigEngine.sh wp-query-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, any time during the configuration process or at runtime to query an overview of the currently defined attributes. This task creates the `availableAttributes.html` report, located in the `wp_profile_root/ConfigEngine/log` directory. The report contains one table that lists the available attributes for Users (**PersonAccount**) and one table that lists the available attributes for Groups. For each configured repository there is a column that indicates if the attribute is flagged as unsupported or if the attribute is mapped to a different LDAP attribute. **Note:** This task does not validate the existence of attributes in the LDAP schema.

**Parent topic:** Adapting the attribute configuration

**Next topic:** Adding attributes on HP-UX in a clustered environment
Adding attributes on HP-UX in a clustered environment

The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM® WebSphere® Portal and your user registry.

Perform the following steps to add new attributes to your user registry:

1. Install the required Enterprise Archive (.ear) file on WebSphere Application Server. Select the appropriate task to install the .ear file:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone environment</td>
<td>Perform the following steps to install the .ear file in a stand-alone environment: Open a command prompt. Navigate to the \wp_profile_root\ConfigEngine directory. Run the ./ConfigEngine.sh wp-la-install-ear -DWasPassword=password task.</td>
</tr>
<tr>
<td>Clustered environment</td>
<td>Perform the following steps to install the .ear file in a clustered environment: Open a command prompt. Navigate to the \wp_profile_root\ConfigEngine directory on the primary node. Run the ./ConfigEngine.sh wp-la-install-ear -DWasPassword=dmgr_password -DServerName=dmgr -DNodeName=node_name task. Where node_name is the name of the node where the deployment manager resides; you can find the node_name value in the WebSphere Application Server Administrative Console under System administrator &gt; Deployment Manager &gt; Runtime tab &gt; General Properties &gt; Node Name.</td>
</tr>
</tbody>
</table>

2. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

3. Use a text editor to open the wkplc.properties file, located in the \wp_profile_root\ConfigEngine\properties directory.

4. Enter a value for the following required parameters in the wkplc.properties file under the VMM Property Extension Properties heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.

   - la.providerURL
   - la.propertyName
   - la.entityTypes
   - la.dataType
   - la.multiValued

5. Save your changes to the wkplc.properties file.
6. Run the `./ConfigEngine.sh wp-add-property -DWasPassword=password` task to add the attribute to the user registry.

   **Note:** This task performs an EJB call to WebSphere Application Server, which must authenticate against WebSphere Application Server. Depending on the configuration in the `sas.client.props` file, you may receive a popup window or a command line prompt asking for user identity and password. Enter the WebSphere Application Server user ID and password.

   **Remember:** If you have multiple properties to add, repeat all steps, except for the `wp-la-install-ear` task, until all new attributes are added.

7. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

   If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See Enabling LDAP security after cluster creation for instructions.

**Parent topic:** Adapting the attribute configuration

**Previous topic:** Querying the defined attributes on HP-UX

**Next topic:** Mapping attributes on HP-UX in a clustered environment

**Related tasks**

- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation
Mapping attributes on HP-UX in a clustered environment

After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

Perform the following steps to map attributes between WebSphere Portal and your LDAP server; if you have multiple LDAP servers, you will need to perform these steps for each LDAP server:

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

2. Enter a value for one of the following sets of parameters in the wkplc.properties file to identify your LDAP server:
   - **Note:** Make sure you use the same values you used to configure your LDAP server.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>The following parameters are found under the LDAP attribute configuration heading: <strong>Note:</strong> See the properties file for specific information about the required parameters and for advanced parameters.</td>
</tr>
<tr>
<td></td>
<td>standalone.ldap.idstandalone.ldap.hoststandalone.ldap.portstandalone.ldap.sslEnabledstandalone.ldap.bindDNstandalone.ldap.bindPasswordstandalone.ldap.baseDN</td>
</tr>
<tr>
<td>Federated</td>
<td>The following parameters are found under the VMM Federated repository properties heading: <strong>Note:</strong> See the properties file for specific information about the required parameters and for advanced parameters.</td>
</tr>
<tr>
<td></td>
<td>federated.ldap.idfederated.ldap.hostfederated.ldap.portfederated.ldap.sslEnabledfederated.ldap.bindDNfederated.ldap.bindPasswordfederated.ldap.baseDN</td>
</tr>
</tbody>
</table>

3. Run one of the following tasks to check that all defined attributes are available in the configured LDAP user registry:

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory.</td>
</tr>
<tr>
<td>Federated</td>
<td>./ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory.</td>
</tr>
</tbody>
</table>

4. Open the ConfigTrace.log file, located in the wp_profile_root/ConfigEngine/log directory, to review the following output for the PersonAccount and Group entity type:
- The following attributes are defined in WebSphere Portal but not in the LDAP server
  - This list contains all attributes that are defined in WebSphere Portal but not available in the LDAP. Flag attributes that you do not plan to use in WebSphere Portal as unsupported. Map the attributes that you plan to use to the attributes that exist in the LDAP; you must also map the `uid`, `cn`, `firstName`, `sn`, `preferredLanguage`, and `ibm-primaryEmail` attributes if they are contained in the list.

- The following attributes are flagged as required in the LDAP server but not in WebSphere Portal
  - This list contains all attributes that are defined as "MUST" in the LDAP server but not as required in WebSphere Portal. You should flag these attributes as required within WebSphere Portal; see the step below about flagging an attribute as either unsupported or required.

- The following attributes have a different type in WebSphere Portal and in the LDAP server
  - This list contains all attributes that WebSphere Portal might ignore because the data type within WebSphere Portal and within the LDAP server do not match.

5. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

6. Enter a value for one of the following sets of parameters in the `wkplc.properties` file to correct any issues found in the config trace file: Table 3. Parameters to define in the `wkplc.properties` file to correct any issues found in the config trace file.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Parameters</th>
</tr>
</thead>
</table>
| Stand-alone     | The following parameters are found under the LDAP attribute configuration heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters. `standalone.id=standalone.ldap.attributes.nonSupported.standalone.ldap.attributes.nonSupported.deletestandalone.ldap.attributes.mapping.ldapName=standalone.ldap.attributes.mapping.portalName=standalone.ldap.attributes.mapping.entityTypes=PersonAccount, Group` For example, the following values will flag `certificate` and `members` as unsupported attributes and will map `ibm-primaryEmail` to `mail` and `ibm-jobTitle` to `title` for both the `PersonAccount` and `Group` entity types. `standalone.id=standalone.ldap.attributes.nonSupported=certificate, members` `standalone.id=standalone.ldap.attributes.nonSupported.delete=standalone.ldap.attributes.mapping.ldapName=mail, title` `standalone.id=standalone.ldap.attributes.mapping.portalName=ibm-primaryEmail, ibm-jobTitle` `standalone.id=standalone.ldap.attributes.mapping.entityTypes=PersonAccount, Group`
7. Save your changes to the wkplc.properties file.

8. Run one of the following tasks to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between WebSphere Portal and the LDAP user registry:

   Table 4. Task to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between Portal and the LDAP user registry.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>./ConfigEngine.sh wp-update-standalone-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory</td>
</tr>
<tr>
<td>Federated</td>
<td>./ConfigEngine.sh wp-update-federated-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory</td>
</tr>
</tbody>
</table>

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

10. Optional: Perform the following steps to flag an attribute as either unsupported or required for the entire WebSphere Portal environment instead of just for the specified LDAP:
   A. Enter a value for the following required parameters in the wkplc.properties file: Note: See the properties file for specific information about the required parameters and for advanced parameters.
      - `user.attributes.required`
      - `user.attributes.nonsupported`
   B. Save your changes to the wkplc.properties file.
   C. Run the `./ConfigEngine.sh wp-update-attribute-config -DWasPassword=password` task, from the wp_profile_root/ConfigEngine directory.
   D. Stop and restart all necessary servers to propagate your changes.

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See Enabling LDAP security after cluster creation for instructions.
Related tasks
Starting and stopping servers, deployment managers, and node agents
Enabling LDAP security after cluster creation
Removing attributes

Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

Perform the following steps to remove an attribute from your database:

**Important:** Do not remove attributes that have already been populated with user values because this can cause database inconsistencies.

**Cluster note:** In a clustered environment, perform the following steps on the deployment manager and then resynch the nodes.

1. Perform the following steps to remove an attribute stored in a property extension database; this is an attribute that was added using the `wp-add-la-property` task.
   A. Open the tool you use to edit your database.
   B. Verify that your attribute name is available in the LAPROP table.
   C. Delete the required attributes from the LAPROP table.
   D. Open the `wimxmlextension.xml` file, located in the `wp_profile_root/config(cells/cellname/wim/model` directory.
   E. Locate and delete the `propertySchema` definition for the attributes that you deleted from the LAPROP table; for example:
      ```xml
      <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String"
                      multiValued="true" propertyName="attribute_name">
        <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
      </wim:propertySchema>
      ```
   F. Save your changes to the `wimxmlextension.xml` file.
   G. Open the `wimconfig.xml` file, located in the `wp_profile_root/config(cells/cellname/wim/config` directory.
   H. Locate and delete the `propertiesNotSupported` definitions for the attributes that you deleted from the LAPROP table; for example:
      ```xml
      <config:propertiesNotSupported name="attribute_name">
      </config:propertiesNotSupported>
      ```
   I. Save your changes to the `wimconfig.xml` file.
   J. Stop and restart the server1 and WebSphere_Portal servers from the `wp_profile_root/bin` directory.

2. Perform the following steps to remove an attribute that is not stored in a property extension database:
   A. Open the `wimxmlextension.xml` file.
   B. Locate and delete the `propertySchema` definition for the attributes you previously added:
      ```xml
      <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String"
                      multiValued="true" propertyName="attribute_name">
        <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
      </wim:propertySchema>
      ```
   C. Save your changes to the `wimxmlextension.xml` file.
   D. Open the `wimconfig.xml` file.
   E. Locate and delete the stanza that corresponds to the custom attribute you deleted from the `wimextension.xml` file; for example:
      ```xml
      <config:attributes name="attribute_name" propertyName="property_name">
        <config:entityTypes>PersonAccount</config:entityTypes>
      </config:attributes>
      ```
   F. Save your changes to the `wimconfig.xml` file.
G. Stop and restart the server1 and WebSphere_Portal servers.

**Parent topic:** Adapting the attribute configuration

**Previous topic:** Mapping attributes on HP-UX in a clustered environment
Preparing additional nodes on HP-UX

After installing and configuring your primary node, you can create your secondary nodes. You must install IBM® WebSphere® Portal on each node and then configure the node to access the database and user registry before adding it to the cluster.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on HP-UX
  - Preparing your HP-UX operating system
  - Preparing the primary node on HP-UX
  - Choosing the type of cluster to create on HP-UX
  - Installing a secondary node after migrating from previous version

Perform the following tasks to prepare your secondary node:

1. **Installing WebSphere Portal on HP-UX on the additional nodes**
   - Install IBM WebSphere Portal on your secondary nodes to create a highly available and scalable environment.

2. **Choosing the type of additional node to create on HP-UX**
   - If you created a static cluster using the IBM WebSphere Application Server Network Deployment, add additional static nodes to the cluster. If you created a dynamic cluster using IBM WebSphere Virtual Enterprise, add additional dynamic nodes to the cluster.

3. **Choosing the type of vertical cluster to create on HP-UX**
   - If you are using the IBM WebSphere Application Server Network Deployment, you can add vertical cluster members to your static cluster. If you are using IBM WebSphere Virtual Enterprise, you can add vertical cluster members to your dynamic cluster.

**Parent topic:** Setting up a cluster on HP-UX

**Previous topic:** Configuring WebSphere Portal to use a user registry on HP-UX in a clustered environment

**Next topic:** Tune your servers
Installing WebSphere Portal on HP-UX on the additional nodes

Install IBM® WebSphere® Portal on your secondary nodes to create a highly available and scalable environment.

Review the Supported hardware and software requirements for your operating system and ensure that all hardware and software matches the minimum product level before installing WebSphere Portal. Review Installation methods, options, and sources before installing to determine your installation source location and installation method. **Note:** If you plan to use IBM WebSphere Application Server Version 7.0, you must purchase and install it before installing WebSphere Portal on the existing WebSphere Application Server Version 7.0 instance.

Perform the following steps to install WebSphere Portal:

1. Type `ping yourserver.yourcompany.com` on a command line to verify that your fully qualified host name is properly configured.
2. If you are installing with an existing WebSphere Application Server instance, ensure it is installed at the supported level and has the following features installed:
   - **Restriction:** Installing WebSphere Portal Version 6.1.0.1 or higher on an existing WebSphere Application Server Version 7.0 instance where WebSphere Portal Version 7.0 was installed is unsupported.
   - **Important:** Besides ensuring that the existing WebSphere Application Server instance is at the supported level, you will also need to ensure that Apache Derby is installed at the supported level before installing WebSphere Portal. See WebSphere Portal detailed system requirements for supported levels.

   - Application Server
   - Administration
     - Scripted Administration
     - Administrative Console
   - Ant and Deployment Tools
     - Deploy Tool
     - Ant Utilities

3. Optional: Perform the following steps if you want to install WebSphere Portal using a non-root user:
   - **Restriction:** Although it is possible to install WebSphere Application Server as a non-root user, there are limitations to this option that can affect WebSphere Portal functionality, which is why you should install WebSphere Application Server as a root user.
   A. Install the current supported version of WebSphere Application Server as a root user.
   B. Open a command prompt and perform the following steps to create a non-root user and to change ownership:
      1. Use the appropriate system commands to create a new group, to create a new user, and to add the new user to the new group.
      2. Run the following tasks to change the rights of the non-root user:
         ```
         chmod -R g+rwX /opt/IBM
         chgrp -R group_name /opt/IBM
         chmod -R g+wr /tmp
         chgrp -R group_name /tmp
         chmod -R g+wr /var/tmp
         ```
C. Launch the installation program per the next step. During the installation, select the existing WebSphere Application Server using the non-root user and set the installation location to a unique location under the path where non-root permissions were granted.

4. Choose one of the following installation commands based on the installation method you want to use to install WebSphere Portal; see Installation Methods for more information:

   **Advance installation parameters:** To customize your installation, you can add various parameters to your installation commands; for example, you can specify your port information. See Advanced installation parameters for information.

   **Restriction:** When defining your installation path, the **ONLY** supported characters are ASCII letters [A-Z and a-z], numbers [0-9], slashes [/ or \, depending on your operating system], and the dash [-].

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Graphical user interface</strong></td>
<td>./install.sh</td>
</tr>
<tr>
<td><strong>Console mode</strong></td>
<td>./install.sh -console from the PortalExpress subdirectory</td>
</tr>
<tr>
<td><strong>Silent Install</strong></td>
<td>./install.sh -options &quot;path_to_file/response_filename&quot;, where path_to_file is the full path to the response file and response_filename is the name of the file. A sample install response file (installresponse.txt) and a sample uninstall response file (uninstallresponse.txt) are located in the setup CD root directory. <strong>Important:</strong> Do not place the response file in a path that contains a space and do not put a space in the file name.</td>
</tr>
</tbody>
</table>

5. Verify your installation was successful; access WebSphere Portal using the http://yourserver:yourport/wps/portal format. Run the ./ConfigEngine.sh list-server-ports -DWasPassword=password task from the wp_profile_root/ConfigEngine directory to generate the wp_profile_root/ConfigEngine/log/wp_PortMatrix.txt file that lists WebSphere Portal ports for your installation.

6. Optional: Perform the following steps if you need to change the ports for WebSphere Application Server or WebSphere Portal:

   **Note:** The starting port parameter is required for a successful completion of the modify-ports-by-startport task.

   Once you specify a start port, this port becomes the base for assigning port values. The code increments this value as each port is assigned, which means that the WebSphere Portal ports will be assigned incrementally starting with the port defined with the modify-ports-by-startport task.

A. Stop the server1 and WebSphere_Portal servers.
B. Run one of the following commands for each server you need to change:

<table>
<thead>
<tr>
<th>Method</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting port number</td>
<td>./ConfigEngine.sh modify-ports-by-startport -DWasPassword=password -DModifyPortsServer=servername -DStartPort=starting port number</td>
</tr>
</tbody>
</table>
C. Restart the server1 and WebSphere_Portal servers.

7. To prevent Out of Memory errors, perform the following steps to set the MaxPermSize:

Tip: If Maximum Heap Size is set to 2048 M or higher, set the MaxPermSize to a quarter of the value entered for the Maximum Heap Size; for example, if your Maximum Heap Size is 3000 M, set MaxPermSize to 750 M. If your Maximum Heap Size is less than 2048 M, set MaxPermSize to 512 M.

A. Log in to the WebSphere Application Server Administration Console.

B. Click Servers > Server Types > WebSphere application servers > WebSphere_Portal.


D. In the Generic JVM arguments field, change the MaxPermSize value to -XX:MaxPermSize=numeric value, where numeric value is a quarter of the value entered for the Maximum Heap Size. Important: If MaxPermSize does not exist in the Generic JVM arguments field, add it to the field but do not replace existing information in the Generic JVM arguments field with the MaxPermSize information.

E. Click OK to save your changes.

F. Click Save to save your changes to the master configuration.

G. Log out of the WebSphere Application Server Administration Console.

H. Restart the server1 and WebSphere_Portal servers.

8. Perform the following steps if you migrated from a previous version of WebSphere Portal:

A. Log on to the WebSphere Application Server Administrative Console.


C. Click WP Identification.

D. Click Custom properties.

E. If applicable, note the location of the LpidToGupidMapping.properties file.

F. If the LpidToGupidMapping.properties file exists on the primary node, copy the file to the same location on the secondary node.

**Parent topic:** Preparing additional nodes on HP-UX

**Next topic:** Choosing the type of additional node to create on HP-UX

**Related concepts**

Installation methods
IBM Support Assistant Lite for WebSphere Portal

Related reference
Advanced installation parameters
Choosing the type of additional node to create on HP-UX

If you created a static cluster using the IBM® WebSphere® Application Server Network Deployment, add additional static nodes to the cluster. If you created a dynamic cluster using IBM WebSphere Virtual Enterprise, add additional dynamic nodes to the cluster.

- **Prerequisites**
  - Installing WebSphere Portal on HP-UX on the additional nodes

Choose one of the following options to add additional nodes to your cluster:

- **Adding additional nodes to the static cluster on HP-UX**
  After installing IBM WebSphere Portal on your secondary node, you must add the node to the cluster to create a highly available environment.

- **Adding an additional node to an existing dynamic cluster**
  After creating your dynamic cluster, you can add additional nodes to expand the capacity of the dynamic cluster. Install and configure the additional node, then add it to the existing dynamic cluster node group, and then add it to the existing IBM WebSphere Virtual Enterprise dynamic cluster.

**Parent topic:** Preparing additional nodes on HP-UX

**Previous topic:** Installing WebSphere Portal on HP-UX on the additional nodes

**Next topic:** Choosing the type of vertical cluster to create on HP-UX
Adding additional nodes to the static cluster on HP-UX

After installing IBM® WebSphere® Portal on your secondary node, you must add the node to the cluster to create a highly available environment.

Perform the following steps to add the secondary node to the cluster:

1. Perform the following steps on the secondary node to access your database server: **Note:** All properties files are located in the `wp_profile_root/ConfigEngine/properties` directory.
   A. For JDBC Type 2 connections only, ensure that the database client software is installed on the secondary node using the same settings as the primary node and that you can connect to the remote database. It is not necessary to create databases or users when configuring secondary nodes. **DB2 users:** When using DB2, you must catalog the TCP/IP node and all databases used on each secondary node. For more information refer to Preparing DB2.
   B. For JDBC Type 4 connections only, copy the JDBC Type 4 jar files to the secondary node from the DB2 server.
   C. Set all database properties in the `wkplc_dbtype.properties` and `wkplc_comp.properties` files with the appropriate values from the primary node.

2. Run the following tasks to validate your database settings:
   A. `./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password`
   B. `./ConfigEngine.sh validate-database-connection -DWasPassword=password -DTransferDomainList=release,customization,community,jcr,feedback,likeminds`

3. Stop the server1 and WebSphere_Portal servers on the secondary node and ensure that the following parameters are set correctly in the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory of the secondary node: **Note:** Although you can add these parameters directly to any task that you run while creating your cluster, you may want to add them to the properties file while creating your cluster and then remove them when you are finished to keep your environment secure.
   A. Verify that `WasUserid` is set to your deployment manager administrator ID.
   B. Verify that `WasPassword` is set to your deployment manager administrator password.
   C. Verify that `PortalAdminId` is set to your WebSphere Portal administrator ID.
   D. Verify that `PortalAdminPwd` is set to your WebSphere Portal password.
   E. Verify that `WasRemoteHostName` is set to the fully qualified host name of the deployment manager.
   F. Verify that `WasSoapPort` is set to the SOAP port that the deployment manager is using; the default value is 8879.
   G. Verify that `PrimaryNode` is set to `false`.
   H. Verify that `ClusterName` is set to the primary node’s `ClusterName`.

4. Run the `./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=dmgr_password` task, from the `wp_profile_root/ConfigEngine` directory of the secondary node.

5. Optional: If you configured a database user registry or a property extension (lookaside) database, run the following task to set up access to the database drivers:
   **Note:** You will also need to run this task if you migrated the primary node from a release prior to Version 6.1.0, even if you did not configure a database user registry or a property extension (lookaside) database.
   A. Set the property value for `federated.db.DbType` if using a database user registry or set the property value for `la.DbType` if using a property extension database in the `wkplc.properties` file.
   **Note:** If you migrated the primary node from a version prior to Version 6.1.0 and you are not using a database user...
registry or a property extension database, set the property value for `federated.dbDbType` to the same value that is in the `wkplc.properties` file on the primary node.

B. Run the `.ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=dmgr_password -DbDomain=localhost -DVmmNodeName=node_name -DbDbType.NodeDbLibrary=local full path of the database jars` task from the `wp_profile_root/ConfigEngine` directory to create the variable used to access the VMM database jars.

**Note:** `VmmNodeName` is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The `db_type` in `db_type.NodeDbLibrary` should be set to the type of database you are using, for example `db2`.

6. Run the `.ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=dmgr_password` task. **Note:** If you want to customize the WebSphere Portal server name (WebSphere_Portal) for your additional node, set the `ServerName` parameter in the `wkplc.properties` file after running the `cluster-node-config-post-federation` task.

7. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell's user registry. Run the `.ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task from the `wp_profile_root/ConfigEngine` directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

**Attention:** The password value for `-DWasPassword` is the Deployment Manager administrative password. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

**Fast path:** If the value for `newAdminGroupId` contains a space; for example `Software Group`, open the `wkplc.properties` file and add the values for `newAdminId`, `newAdminPw`, and `newAdminGroupId`. Save your changes and run the `ConfigEngine.bat wp-change-portal-admin-user -DWasPassword=dmgr_password` task.

**Important:** If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user -DnewAdminPw` is an optional parameter to update the Administrative password in the `wkplc.properties` file if required.

8. Optional: Open the `wkplc.properties` file and change the `ServerName` parameter from the default WebSphere_Portal_nodename value to a value that meets your business needs. Do not change the parameter to WebSphere_Portal as this is the primary node value.

9. Run the `.ConfigEngine.sh cluster-node-config-cluster-setup -DWasPassword=dmgr_password` task to add the node to your cluster.

10. Wait at least 30 minutes before stopping and restarting the deployment manager because the node agent runs a background thread to expand all EAR files into their target directories. **Note:** Attempting to restart the deployment manager while the expansion is in process can result in an unusable application. Although the time to complete an EAR expansion is environment-dependent, a 30 minute wait is sufficient for a typical environment.

11. Perform the following steps to access the Web Content Management content through an external Web server:

A. Log on to the deployment manager administrative console.
B. Select `Environment > WebSphere Variables`.
C. From the `Scope` drop-down menu, select the `Node=nodename, Server=servername` option to narrow the scope of the listed variables, where `nodename` is the node that contains the application server.
D. Update the **WCM_HOST** variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.

E. Update the **WCM_PORT** variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.

F. Perform the following steps to synchronize the node with the deployment manager:
   1. Select **System Administration > Nodes**.
   2. Select the node that you want to synchronize from the list.
   3. Click **Full Resynchronize**.

G. Log off of the deployment manager administrative console.

12. If you entered passwords in any of the properties files while creating your cluster, you should remove them for security purposes. See "Deleting passwords from properties files" under Related tasks for information.

Parent topic: Choosing the type of additional node to create on HP-UX

Related tasks
Deleting passwords from properties files
Adding an additional node to an existing dynamic cluster

After creating your dynamic cluster, you can add additional nodes to expand the capacity of the dynamic cluster. Install and configure the additional node, then add it to the existing dynamic cluster node group, and then add it to the existing IBM® WebSphere® Virtual Enterprise dynamic cluster.

On all WebSphere Portal nodes, install WebSphere Virtual Enterprise and augment the wp_profile profile. See Installing and configuring the product for information about installing WebSphere Virtual Enterprise. See Augmenting profiles for information about augmenting the Deployment Manager and wp_profile profiles.

**Prerequisites**

- PK97393 for all WebSphere Application Server Versions prior to 6.1.0.5
- Installing and configuring the product:
- Augmenting profiles

Perform the following steps to add an additional node to an existing WebSphere Virtual Enterprise dynamic cluster:

1. Perform the following steps on the secondary node to access your database server. **Note:** All properties files are located in the \wp_profile_root\ConfigEngine\properties directory.
   A. For JDBC Type 2 connections only, ensure that the database client software is installed on the secondary node using the same settings as the primary node and that you can connect to the remote database. It is not necessary to create databases or users when configuring secondary nodes. **DB2 users:** When using DB2, you must catalog the TCP/IP node and all databases used on each secondary node. For more information refer to Preparing DB2.
   B. For JDBC Type 4 connections only, copy the JDBC Type 4 jar files to the secondary node from the DB2 server.
   C. Set all database properties in the \wkplc_dbtype.properties and \wkplc_comp.properties files with the appropriate values from the primary node.

2. Run the following tasks to validate your database settings:
   A. `./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password`
   B. `./ConfigEngine.sh validate-database-connection -DWasPassword=password -DTransferDomainList=release,customization,community,jcr,feedback,likeminds`

3. Stop the server1 and WebSphere_Portal servers on the secondary node and ensure that the following parameters are set correctly in the wkplc.properties file, located in the \wp_profile_root\ConfigEngine\properties directory of the secondary node. **Note:** Although you can add these parameters directly to any task that you run while creating your cluster, you may want to add them to the properties file while creating your cluster and then remove them when you are finished to keep your environment secure.
   A. Verify that **WasUserid** is set to your deployment manager administrator ID.
   B. Verify that **WasPassword** is set to your deployment manager administrator password.
   C. Verify that **PortalAdminId** is set to your WebSphere Portal administrator ID.
   D. Verify that **PortalAdminPwd** is set to your WebSphere Portal password.
   E. Verify that **WasRemoteHostName** is set to the fully qualified host name of the deployment manager.
   F. Verify that **WasSoapPort** is set to the SOAP port that the deployment manager is using; the default value is 8879.
   G. Verify that **PrimaryNode** is set to false.
H. Verify that `ClusterName` is set to the primary node’s `ClusterName`.

4. Run the `./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=dmgr_password` task, from the `wp_profile_root/ConfigEngine` directory of the secondary node.

5. Optional: If you configured a database user registry or a property extension (lookaside) database, run the following task to set up access to the database drivers:

   **Note:** You will also need to run this task if you migrated the primary node from a release prior to Version 6.1.0, even if you did not configure a database user registry or a property extension (lookaside) database.

   A. Set the property value for `federated.db.DbType` if using a database user registry or set the property value for `la.DbType` if using a property extension database in the `wkplc.properties` file.

      **Note:** If you migrated the primary node from a version prior to Version 6.1.0 and you are not using a database user registry or a property extension database, set the property value for `federated.db.DbType` to the same value that is in the `wkplc.properties` file on the primary node.

   B. Run the `./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=dmgr_password -DDbDomain=la|federated.db -DVmmNodeName=node_name -Db_type.NodeDbLibrary=local full path of the database jars` task from the `wp_profile_root/ConfigEngine` directory to create the variable used to access the VMM database jars.

      **Note:** `VmmNodeName` is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The `db_type` in `db_type.NodeDbLibrary` should be set to the type of database you are using, for example `db2`.

6. Run the `./ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=dmgr_password` task. **Note:** If you want to customize the WebSphere Portal server name (WebSphere_Portal) for your additional node, set the `ServerName` parameter in the `wkplc.properties` file after running the `cluster-node-config-post-federation` task.

7. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell’s user registry. Run the `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task, from the `wp_profile_root/ConfigEngine` directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

   **Attention:** The password value for `-DWasPassword` is the Deployment Manager administrative password.

   **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

   **Fast path:** If the value for `newAdminGroupId` contains a space; for example `Software Group`, open the `wkplc.properties` file and add the values for `newAdminId`, `newAdminPw`, and `newAdminGroupId`. Save your changes and run the `ConfigEngine.bat wp-change-portal-admin-user -DWasPassword=dmgr_password` task.

   **Important:** If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user` task until after the `cluster-node-config-cluster-setup` (static cluster) or `cluster-node-config-dynamic-cluster-setup` (dynamic cluster) task completes. After running the `wp-change-portal-admin-user` task, start or restart the WebSphere_Portal server to use the updated administrator user ID.

   **Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the `wp-change-portal-admin-user` task. `-DnewAdminPw` is an optional parameter to update the Administrative password in the `wkplc.properties` file if required.

8. Log on to the deployment manager administrative console.

9. Perform the following steps to add members to the node group:

   A. Click `System administration > Node groups`. 
B. Click on the name of the node group that you want to add members to.
C. Click **Node group members** under Additional Properties.
D. Click **Add**.
E. Select the additional node you want to add into the node group and then click **Add**.
F. Click the **Save** link to save your changes to the master configuration.

10. Wait at least 30 minutes before stopping and restarting the deployment manager because the node agent runs a background thread to expand all EAR files into their target directories. **Note:** Attempting to restart the deployment manager while the expansion is in process can result in an unusable application. Although the time to complete an EAR expansion is environment-dependent, a 30 minute wait is sufficient for a typical environment.

11. Define or verify the following parameters in the wkplc.properties file, located in the **wp_profile_root**/ConfigEngine/properties directory:

   A. Set **ClusterName** to the name of the existing dynamic cluster.
   B. Verify that **CellName** is set to the deployment manager cell.
   C. Verify that **NodeName** is set to the local WebSphere Portal node.
   D. Set **ServerName** to the server that will be used for the dynamic cluster member on this node. **Note:** Log on to the deployment manager administrative console and click **Dynamic Clusters > PortalCluster > Dynamic cluster members** to find the name of the server used for the dynamic cluster.
   E. Verify that **PrimaryNode** is set to **false** because this is an additional node.

12. Run the ./ConfigEngine.sh cluster-node-config-dynamic-cluster-setup -DWasPassword=dmgr_password task to add the new member to the existing dynamic cluster. **Note:** This task may display a warning message since the cluster already exists but it will proceed to create the new dynamic cluster; therefore, the warning can be ignored.

13. Perform the following steps to start the cluster member:

   A. Log on to the Deployment Manager administrative console.
   B. Navigate to **Servers > Dynamic clusters > cluster name > Dynamic cluster members**.
   C. Select the cluster member and click **Start**.
   D. Log off of the deployment manager administrative console.

14. Perform the following steps to access the Web Content Management content through an external Web server:

   A. Log on to the deployment manager administrative console.
   B. Select **Environment > WebSphere Variables**.
   C. From the **Scope** drop-down menu, select the **Node=nodename, Server=servername** option to narrow the scope of the listed variables, where **Node=nodename** is the node that contains the application server.
   D. Update the **WCM_HOST** variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.
   E. Update the **WCM_PORT** variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.
   F. Perform the following steps to synchronize the node with the deployment manager:
      1. Select **System Administration > Nodes**.
      2. Select the node that you want to synchronize from the list.
      3. Click **Full Resynchronize**.
   G. Log off of the deployment manager administrative console.

15. Run the following tasks to propagate your changes. **Note:** **WebSphere_Portal_nodename** is the name of the node’s WebSphere Portal server; but if you customized the server name, the default name changes. If you are uncertain of the server name, run the **serverstatus -all** task to get a list of the server names and their status.

   A. ./stopManager.sh -username admin_userid -password admin_password, from the dmgr_profile_root\bin directory
   B. ./stopNode.sh -username admin_userid -password admin_password, from the wp_profile_root/bin directory
   C. ./stopServer.sh WebSphere_Portal_nodename -username admin_userid -password admin_password
D. ./startManager.sh, from the dmgr_profile_root/bin directory
E. ./startNode.sh, from the wp_profile_root/bin directory
F. ./startServer.sh WebSphere_Portal_nodename

16. If you entered passwords in any of the properties files while creating your cluster, you should remove them for security purposes. See "Deleting passwords from properties files" under Related tasks for information.

**Parent topic:** Choosing the type of additional node to create on HP-UX

**Related tasks**
Deleting passwords from properties files
Choosing the type of vertical cluster to create on HP-UX

If you are using the IBM® WebSphere® Application Server Network Deployment, you can add vertical cluster members to your static cluster. If you are using IBM WebSphere Virtual Enterprise, you can add vertical cluster members to your dynamic cluster.

- **Prerequisites**
  - Installing WebSphere Portal on HP-UX on the additional nodes

Choose one of the following options to add vertical cluster members to your cluster.

- **Adding vertical cluster members to a static cluster on HP-UX**
  You can add vertical cluster members to share the workload demands of your cluster across multiple members running on the same physical machine.

- **Adding vertical cluster members to a dynamic cluster on HP-UX**
  You can add vertical cluster members to share the workload demands of your cluster across multiple members running on the same physical machine.

**Parent topic:** Preparing additional nodes on HP-UX

**Previous topic:** Choosing the type of additional node to create on HP-UX
Adding vertical cluster members to a static cluster on HP-UX

You can add vertical cluster members to share the workload demands of your cluster across multiple members running on the same physical machine.

Perform the following steps to add a vertical cluster member to your clustered environment:

1. Log into the deployment manager administrative console.
2. Depending on your version of WebSphere Application Server, make the appropriate selection from the navigation:
   - For WebSphere Application Server Version 6.1: Click Servers > Clusters in the console navigation tree, select the cluster name, and then click Cluster members from the list of additional properties.
   - For WebSphere Application Server Version 7.0: Click Servers > Clusters > WebSphere application server clusters > cluster_name > Cluster members.
   - Click New to create the cluster member.
   - Define the name of cluster member. Note: Do not use spaces in the cluster member name.
   - Select an existing node where IBM® WebSphere® Portal is installed.
   - Check the box Generate Unique HTTP Ports.
   - Click Add Member and then click Next to view the summary.
   - Important: You must update the virtual host entries for the new port created when adding a cluster member. You can do this by updating the default_host virtual host in the administrative console and adding a new alias entry for the port number (use an "*" wildcard character for the host name). See Configuring virtual hosts for information.
   - Click Finish, and save the changes.
3. Perform the following steps to enable cache replication:
   - From the deployment manager Administrative Console, navigate to Servers > Application servers and then click the new vertical cluster member(s). If using WebSphere Application Server Version 7.0, navigate to Servers > Server Types > WebSphere application servers and then click the new server cluster members.
   - Click Dynamic cache service under Container services.
   - Change Cache size to 3000 entries.
   - Check the Enable cache replication check box.
   - Select NOT_SHARED from the Replication type drop-down menu.
   - Click OK.
   - Click Save to save your changes to the master configuration.
4. Perform the following steps on each vertical cluster member to clean up the server-scoped resources, caches, and resource providers:
   - Run the ./ConfigEngine.sh cluster-node-config-vertical-cluster-setup -DServerName=unique vertical cluster servername -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, where unique vertical cluster servername is the name you specified when you created the cluster member.
   - Restart the vertical cluster member referenced in the cluster-node-config-vertical-cluster-setup task.
7. Perform the following steps to access the Web Content Management content through an external Web server:
   A. Log on to the deployment manager administrative console.
   B. Select Environment > WebSphere Variables.
   C. From the Scope drop-down menu, select the Node-nodename, Server-servername option to narrow the scope of the listed variables, where Node-nodename is the node that contains the application server.
   D. Update the WCM_HOST variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.
   E. Update the WCM_PORT variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.
   F. Perform the following steps to synchronize the node with the deployment manager:
      1. Select System Administration > Nodes.
      2. Select the node that you want to synchronize from the list.
      3. Click Full Resynchronize.
   G. Log off of the deployment manager administrative console.
8. Save your changes and resynchronize the nodes.
   A. In the administrative console for the deployment manager, click Save on the task bar, and save your administrative configuration.
   B. Select System Administration > Nodes, select the node from the list, and click Full Resynchronize.
9. Regenerate the Web server plug-in.
   A. Regenerate the Web server plug-in using the deployment manager administrative console.
   B. If you are using a remote Web server, copy the updated plug-in configuration file (plugin-cfg.xml) to the Web server’s plug-in configuration directory.
10. Stop and start the Web server.

Parent topic: Choosing the type of vertical cluster to create on HP-UX
Adding vertical cluster members to a dynamic cluster on HP-UX

You can add vertical cluster members to share the workload demands of your cluster across multiple members running on the same physical machine.

Perform the following steps to add a vertical cluster member to your clustered environment:

1. Open a browser and enter http://DM01:9060/ibm/console in the address bar to access the administrative console on the deployment manager, where DM01 is the deployment manager node or host name. The port number might differ based on your installation.

2. Perform the following steps to allow vertical clusters on your dynamic cluster:
   A. Navigate to Servers > Dynamic clusters and select the appropriate dynamic cluster.
   B. Select the Allow more than one instance to start on the same node check box under Vertical stacking of Instances on node.
   C. Enter a new value in the Number of instances text box to determine the number of vertical cluster members allowed on each node.
   D. Click Apply and then click Save to save the changes to the master configuration.

   The new cluster topology can be viewed from the Servers > Cluster Topology view. If using WebSphere Application Server Version 7.0, the new cluster topology can be viewed from the Servers > Clusters > Cluster Topology view.

   Important: You must update the virtual host entries for the new port created when adding a cluster member. You can do this by updating the default_host virtual host in the administrative console and adding a new alias entry for the port number (use an "*" wildcard character for the host name). See Configuring virtual hosts for information.

3. Perform the following steps to enable cache replication:
   A. From the deployment manager Administrative Console, navigate to Servers > Application servers and then click the new vertical cluster member(s). If using WebSphere Application Server Version 7.0, navigate to Servers > Server Types > WebSphere application servers and then click the new vertical cluster member(s).
   B. Click Dynamic cache service under Container services.
   C. Change Cache size to 3000 entries.
   D. Check the Enable cache replication check box.
   E. Select NOT_SHARED from the Replication type drop-down menu.
   F. Click OK.
   G. Click Save to save your changes to the master configuration.

4. Perform the following steps on each vertical cluster member to clean up the server-scoped resources, caches, and resource providers:
   A. Run the ./ConfigEngine.sh cluster-node-config-vertical-cluster-setup -DServerName=unique vertical cluster servername -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, where unique vertical cluster servername is the name you specified when you created the cluster member.
   B. Restart the vertical cluster member referenced in the cluster-node-config-vertical-cluster-setup task.

5. Perform the following steps to access the Web Content Management content through an external Web server:
   A. Log on to the deployment manager administrative console.
B. Select Environment > WebSphere Variables.
C. From the Scope drop-down menu, select the Node=nodename, Server=servername option to narrow the scope of the listed variables, where Node=nodename is the node that contains the application server.
D. Update the WCM_HOST variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.
E. Update the WCM_PORT variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.
F. Perform the following steps to synchronize the node with the deployment manager:
   1. Select System Administration > Nodes.
   2. Select the node that you want to synchronize from the list.
   3. Click Full Resynchronize.
G. Log off of the deployment manager administrative console.
6. Save your changes and resynchronize the nodes.
   A. In the administrative console for the deployment manager, click Save on the task bar, and save your administrative configuration.
   B. Select System Administration > Nodes, select the node from the list, and click Full Resynchronize.
7. Stop and start the Web server.

Parent topic: Choosing the type of vertical cluster to create on HP-UX
Tune your servers

Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on HP-UX
  - Preparing your HP-UX operating system
  - Preparing the primary node on HP-UX
  - Choosing the type of cluster to create on HP-UX

**Base Portal Tuning Scenarios**

Tuning a WebSphere Portal environment involves tuning and configuring the various systems and components of the environment. The tuning guide provides general concepts and details specifics configuration instructions. Instructions are included for:

- Configuring the application server and the resources defined for that application server
- Determining the cloning strategy for expanding or extending the environment
- Tuning the database(s) and database server
- Tuning the directory server and its database
- Tuning the Web server
- Tuning the operating system and network
- Tuning the WebSphere Portal services

**Parent topic:** Setting up a cluster on HP-UX

**Previous topic:** Preparing additional nodes on HP-UX

**Next topic:** Configuring search in a cluster on HP-UX

**Related information**
- WebSphere Portal and Lotus Web Content Management 6.1.x Performance Tuning Guide
- Tuning a Cluster Environment
- IBM WebSphere Portal Performance Troubleshooting Guide
Configuring search in a cluster on HP-UX

IBM® WebSphere® Portal provides two distinct search capabilities. You can utilize both types of search capabilities in a clustered environment.

**Prerequisites**
- Preparing prerequisite and corequisite software on HP-UX
- Preparing your HP-UX operating system
- Preparing the primary node on HP-UX
- Choosing the type of cluster to create on HP-UX
- Tune your servers

**Configuring Portal Search in a cluster on HP-UX**
To support Portal Search in a clustered environment, you must install and configure search for remote search service on an IBM WebSphere Application Server node that is not part of the IBM WebSphere Portal cluster.

**Configuring JCR search in a cluster on HP-UX**
To enable search in a cluster for content stored in the JCR database, you must configure each server in the cluster to access a shared directory. JCR-based content includes content created with Web Content Management or Personalization.

**Parent topic:** Setting up a cluster on HP-UX  
**Previous topic:** Tune your servers  
**Next topic:** Setting up multiple clusters on HP-UX
Configuring Portal Search in a cluster on HP-UX

To support Portal Search in a clustered environment, you must install and configure search for remote search service on an IBM® WebSphere® Application Server node that is not part of the IBM WebSphere Portal cluster.

To install and configure the search service remotely, perform the following tasks:

1. Install and configure the search service to work remotely, that is, on a remote WebSphere Application Server node which is not part of the portal cluster. You can provide the remote search service either as an EJB or as a Web service via SOAP. Deploy the appropriate EJB or SOAP EAR file on the remote WebSphere Application Server node. For details about how to do this, refer to the WebSphere Application Server documentation.

2. Configure the search portlets for remote search service so that they access the remote server accordingly.

Notes:

1. If you have configured a remote search service for a portal cluster, you need to configure the default location for search collections to a directory on the remote server that has write access.

2. The portal site default search collection is created only once at the first time when an administrator selects the search administration portlet Manage Search. If this occurred before you configure the portlet for remote search, then the default portal site search collection is only available on the primary node of the cluster, but not on the remote server. In this case you need to recreate the portal site collection to make it available for search on all nodes of the cluster.

Parent topic: Configuring search in a cluster on HP-UX

Related concepts

Planning and preparing for Portal Search
Using remote search service

Related tasks

Configuring Portal Search for remote search service
Configuring the Search and Browse portlet for remote search service
Configuring the default location for search collections
Creating or resetting the portal site collection
Configuring a remote search service
Configuring JCR search in a cluster on HP-UX

To enable search in a cluster for content stored in the JCR database, you must configure each server in the cluster to access a shared directory. JCR-based content includes content created with Web Content Management or Personalization.

Create a shared directory called `jcr/search` on a server in the network and ensure that each node in the cluster and the Deployment Manager has network access to the directory.

**Note:** If you are creating content in a clustered environment using the authoring portlet provided with Web Content Management, additional configuration steps are required to enable content created by these content features to be searchable in a cluster.

Perform the following steps on each server in the cluster to configure Search in a clustered environment:

1. Edit the `icm.properties` file, located in the `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm` directory.
2. Change the value of the `jcr.textsearch.indexdirectory` property to the shared directory; for example, `jcr.textsearch.indexdirectory=\\your_server\your_share\jcr\search`. You can specify the shared directory value in one of the following formats: *Table 1. The format for the shared directory.*

<table>
<thead>
<tr>
<th>Format</th>
<th>Shared directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal Naming Convention (UNC) format</td>
<td><code>\\your_server\your_share\jcr\search</code></td>
</tr>
<tr>
<td>Example:</td>
<td><code>\\hostname.example.com\share\jcr\search</code></td>
</tr>
<tr>
<td>Mounted resource format (with forward slashes)</td>
<td><code>/your_share/jcr/search</code> For example: <code>/mnt/jcr/search</code></td>
</tr>
<tr>
<td>Important: This format requires that you mount the shared directory to the local server (for example, through a mapped network drive or a mounted directory). When using the mounted resource format, always use forward slashes instead of back slashes, regardless of the native operating system path format.</td>
<td></td>
</tr>
</tbody>
</table>

3. Required: Perform the following steps to delete the default search collections from the Manage Search portlet:
   A. Log on to WebSphere Portal as an administrator.
   B. Click **Administration > Search Administration > Manage Search**.
   C. Click **Search Collections**.
   D. Click the **Delete Collection** icon for the **Portal Content** search collection.
   E. Click **OK**.
   F. Restart the WebSphere_Portal server.
   G. Go to the Manage Search portlet and confirm that the **Portal Content** search collection was deleted.

**Parent topic:** Configuring search in a cluster on HP-UX
Setting up multiple clusters on HP-UX

The majority of the steps to build another cluster are the same as when building the first cluster in the same cell, with a few exceptions. Basically, the new profile will be designated as the primary profile, using IBM® WebSphere® Portal clustering terminology, and will then be used as the basis for the new cluster definition. This duplicates the build process of the first cluster in the cell. During the federation process, if any applications on this new node already exist in the cell (because they are in use by the first cluster), then Deployment Manager will not allow them to be added. After federation, the applications that already exist in the cell are not mapped to the WebSphere_Portal server on the newly federated node, and thus the existing applications must be remapped to this newly federated server to restore its application list. Therefore, depending on the configuration of the new profile, there will likely be some combination of applications shared with the other existing cluster, and some applications unique to this new profile.

Before attempting alternative approaches for building multiple portal-based clusters within a single cell, please contact IBM.

- Prerequisites
  - Preparing prerequisite and corequisite software on HP-UX
  - Preparing your HP-UX operating system
  - Preparing the primary node on HP-UX
  - Choosing the type of cluster to create on HP-UX
  - Tune your servers

1. Installing multiple clusters in a single cell on HP-UX
   Create a new, independent IBM WebSphere Portal cluster in a cell where a WebSphere Portal cluster already exists.

2. Routing requests across clusters on HP-UX
   The HTTP Server plug-in that comes with IBM WebSphere Application Server is typically used to balance requests for applications across members of the cluster. While an application can be mapped to more than one cluster, automatic plug-in generation does not provide routing or balancing traffic for the same application across multiple clusters. Multiple cluster environments with shared applications therefore cannot rely solely on WebSphere Application Server automatic plug-in generation to be able to route requests using a web server. One option in this scenario is developing a customized method for defining and maintaining the plug-in configuration file used by the Web server to provide for the required routing of user requests.

Parent topic: Setting up a cluster on HP-UX
Previous topic: Configuring search in a cluster on HP-UX
Next topic: Sharing database domains between clusters on HP-UX

|||
Installing multiple clusters in a single cell on HP-UX

Create a new, independent IBM® WebSphere® Portal cluster in a cell where a WebSphere Portal cluster already exists. In the following steps, **Cluster A** will be used to describe the existing cluster. **Portal B** will be used to describe the new server profile that will be the basis for the new cluster definition, **Cluster B**. Perform the following steps to install multiple clusters in a single cell:

1. Upgrade **Cluster A**, including the Deployment Manager node, to the current, supported hardware and software levels and to the current version of IBM WebSphere Portal.
2. Install and configure **Portal B**; see the "Preparing the primary node" topic for the appropriate operating system for details. **Important:** Maintain the same number of data sources with identical names to the **Cluster A** data sources so that data source bindings in the applications can be resolved on every cluster in which they run. If implementing database sharing across the clusters, the above statement refers to both the shared and non-shared domains; all domains should use the same names.
3. Optional: Using the same database user ID and password for each identically named domain/data source will allow the existing JAAS Authentication Aliases to be functional. If unique database user ID and password are required, additional manual configuration is needed to create new JAAS Authentication Aliases for each data source and map these accordingly. On the primary node of **Cluster A**, run the `./ConfigEngine.sh create-alias-multiple-cluster -DauthDomainList=release,jcr -DWasPassword=dmgr_password` task from the `wp_profile_root/ConfigEngine` directory to create the new JAAS Authentication Aliases. where `authDomainList` is set to a list of domains which use unique database user ID and passwords and those domain properties are set correctly in the `wkplc_comp.properties` file, including user ID and password.
4. Optional: If necessary, upgrade **Portal B** to the current cumulative fix.
5. Run the `./ConfigEngine.sh mapped-app-list-create -DWasPassword=password` task from the `wp_profile_root/ConfigEngine` directory to build an inventory list of **Portal B** enterprise applications and portlets.
6. Stop the server1 and WebSphere_Portal servers on **Portal B** and ensure that the following parameters are set correctly in the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory:
   - Set `WasSoapPort` to the deployment manager's port.
   - Set `WasRemoteHostName` to the full host name of the deployment manager.
7. If the Deployment Manager is configured to use a stand-alone LDAP user registry, update the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory, on the primary node with the stand-alone LDAP user registry property values from the Deployment Manager. You can find these settings under the VMM Stand-alone LDAP configuration heading.
   **Important:** Ensure that you set `WasUserid` and `WasPassword` to the Deployment Manager user ID and password.
8. Run the `./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` of the primary node.
9. Run the `./ConfigEngine.sh map-apps-to-server -DWasPassword=password` task to determine which applications from the inventory list are no longer mapped to **Portal B**. The task uses the application profiles already in the cell to restore the mappings. Wait 30 minutes after running this task to allow all EAR files to expand before proceeding to the next step.
10. Perform the following steps to federate **Portal B** into the deployment manager cell:
   A. Ensure that all database parameters are correctly set, including passwords, in the `wkplc_comp.properties` and `wkplc_dbtype.properties` files.
B. Run the "./ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=dmgr_password" task.

C. After running the cluster-node-config-post-federation task, wait at least 30 minutes to allow all EAR files to expand.

D. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell's user registry. Run the "./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid" task, from the wp_profile_root/ConfigEngine directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

Additional parameter for stopped servers: This task verifies the user against a running server instance. If the server is stopped, add the -Dskip.ldap.validation=true parameter to the task to skip the validation.

Important: If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the wp-change-portal-admin-user task until after the cluster-node-config-cluster-setup (static cluster) or cluster-node-config-dynamic-cluster-setup (dynamic cluster) task completes. After running the wp-change-portal-admin-user task, start or restart the WebSphere_Portal server to use the updated administrator user ID.

Note: The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the wp-change-portal-admin-user task. -DnewAdminPw is an optional parameter to update the Administrative password in the wkplc.properties file if required.

E. From the administrative console, click System Administration > Node Agents.

F. Check the box next to the required node agent and then click Restart.

G. Stop and restart the deployment manager.

H. Stop and restart the WebSphere_Portal server on Portal B.

11. Restart the WebSphere_Portal server on Cluster A. Verify that Cluster A is functionally intact by spot checking pages and portlets and then verify that Portal B is functionally intact by spot checking pages and portlets that you deployed into Portal B before it was federated. Any discrepancies or errors should be corrected now before continuing. Note: If Portal B is using a non-default Portal server administrative ID, not wpsadmin, the server will not be functional until the cluster configuration is complete and the Portal administrative ID has been configured to match the Cells security settings.

12. Choose one of the following options to define a cluster using Portal B as the basis:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>

1484
Perform the following steps to define a static cluster using Portal B as the basis:

- Run the `./ConfigEngine.sh cluster-node-config-cluster-setup -DWasPassword=dmgr_password` task.

Configure the cluster to use an external Web server to take advantage of features such as workload management.

Choose one of the following options:

- Configuring a Web server and an application server on separate machines (remote)
- Configuring a Web server and an application server profile on the same machine
- Configuring a Web server and a deployment manager profile on the same machine

**Note:** Start with the step about launching the Plug-ins installation wizard.

Perform the following steps to access the Web Content Management content through an external Web server:

- Log on to the deployment manager administrative console.
- Select Environment > WebSphere Variables. From the Scope drop-down menu, select the Node=nodename, Server=servername option to narrow the scope of the listed variables, where Node=nodename is the node that contains the application server.
- Update the WCM_HOST variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.
- Update the WCM_PORT variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.
- Save your changes and then restart the deployment manager, the node agent(s), server1, and the WebSphere_Portal servers.
- Change the task list settings to point to the cluster instead of a single server if using Portal business process integration. See one of the following sections in the Information Center for information:
  - Cross cell: Adding a BPI-enabled portal server to a managed cell in a cross-cell setup
  - Single cell: Adding a BPI-enabled portal server to a managed cell in a single-cell setup
Dynamic cluster

Perform the following steps to define a dynamic cluster using Portal B as the basis:

Log on to the deployment manager administrative console.

Perform the following steps to create a node group:

1. Click **New**. Type the node group **Name**. Type any information about the node group in the **Description** text box.
2. Click **OK**. Click the **Save** link to save your changes to the master configuration.

Perform the following steps to add members to the node group:

1. Click **System administration > Node groups**. Click on the name of the node group that you want to add members to.
2. Click **Node group members** under **Additional Properties**. Click **Add**. Select the primary node and then click **Add**. Click the **Save** link to save your changes to the master configuration.

Perform the following steps to create a dynamic cluster in the node group:

1. Click **Servers > Dynamic clusters**. Click **New**. Select WebSphere Application Server from the **Server Type** pull-down menu and then click **Next**. Type the cluster name in the **Dynamic cluster name** text box and then click **Next**. Type the same value that you provided for the **ClusterName** parameter in the wkplc.properties file of your primary node.
2. Remove all default membership policies and then click **Subexpression builder**. Enter the following information in the Subexpression builder window:
   - **Logical operator** pull-down menu: **and**
   - **Select operand** pull-down menu: **Nodegroup**
   - **Operator** pull-down menu: **Equals (=)**
   - **Value** text box: Type the nodegroup name you created in the previous step.
3. Click **Generate subexpression**. Click **Append**. Click **Preview membership** to verify that all nodes included in the nodegroup display and then click **Next**. Click the **Create the cluster member using an existing server as a template** radio button and then select the WebSphere Portal server for the primary node from the pull-down menu.
4. Click **Next**. Specify the dynamic cluster properties and then click **Next**. Review the summary page to verify your actions and then click **Finish**.

Define or verify the following parameters in the wkplc.properties file, located in the **wp_profile_root**/ConfigEngine/properties directory:

- **ClusterName** to the name of the new dynamic cluster.
- **CellName** is set to the deployment manager cell.
- **NodeName** is set to the local WebSphere Portal node.
- **ServerName** to the server that will be used for the dynamic cluster member on this node.
- **PrimaryNode** is set to true.

Run the **./ConfigEngine.sh cluster-node-config-dynamic-cluster-setup -DWasPassword=dmgr_password** task from the **wp_profile_root**/ConfigEngine directory to create the dynamic
13. Install any additional nodes to the cell to support additional cluster members for Cluster B identically to the primary node, and then federate as them as secondary nodes and define as cluster members on these nodes. For information about adding additional nodes navigate to Installing WebSphere Portal > Setting up WebSphere Portal > Setting up a clustered production environment. Select Environment > WebSphere Variables. From the Scope drop-down menu, select the Node=nodename, Server=servername option to narrow the scope of the listed variables, where Node=nodename is the node that contains the application server. Update the WCM_HOST variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router. Update the WCM_PORT variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router. Save your changes and then restart the deployment manager, the node agent(s), server1, and the WebSphere Portal servers. If you are using a Web server to connect to the On Demand Router (ODR), configure the web server as a trusted proxy on the ODR. Refer to Configuring a Web server as a trusted proxy server for instructions. Tip: You can also configure the ODR to dynamically update the Web server configuration when changes occur. Refer to Configuring an on demand router to dynamically update the Web server plug-in configuration for instructions. Change the task list settings to point to the cluster instead of a single server if using Portal business process integration. See one of the following sections in the Information Center for information: Cross cell: Adding a BPI-enabled portal server to a managed cell in a cross-cell setup Single cell: Adding a BPI-enabled portal server to a managed cell in a single-cell setup.

14. Restart the server1 and WebSphere_Portal servers on Cluster A and Cluster B.

15. If you entered passwords in any of the properties files while creating your cluster, you should remove them for security purposes. See "Deleting passwords from properties files" under Related tasks for information.

Installation of Cluster B is complete. It is now an independent cluster from Cluster A, which means that Cluster B can have its own configuration, set of end-user portlets, and target community. Any applications that are common between Cluster A and Cluster B are most likely infrastructure or related to administration, and special care needs to be taken to preserve their commonality between clusters and correct maintenance levels.

Parent topic: Setting up multiple clusters on HP-UX
Next topic: Routing requests across clusters on HP-UX
Related tasks
Deleting passwords from properties files
Adding a BPI-enabled portal server to a managed cell in a cross-cell setup
Adding a BPI-enabled portal server to a managed cell in a single-cell setup
Routing requests across clusters on HP-UX

The HTTP Server plug-in that comes with IBM® WebSphere® Application Server is typically used to balance requests for applications across members of the cluster. While an application can be mapped to more than one cluster, automatic plug-in generation does not provide routing or balancing traffic for the same application across multiple clusters. Multiple cluster environments with shared applications therefore cannot rely solely on WebSphere Application Server automatic plug-in generation to be able to route requests using a web server. One option in this scenario is developing a customized method for defining and maintaining the plug-in configuration file used by the Web server to provide for the required routing of user requests.

**Prerequisites**

- Installing multiple clusters in a single cell on HP-UX

An important consideration in a multiple cluster environment is ensuring that all subsequent HTTP requests for an end user are routed to the same cluster that processed the first HTTP request. The WebSphere Portal login processing depends upon preserving this cluster affinity during this initial time until the user has successfully logged in and session cookies maintain affinity. In order to guarantee that affinity is preserved during login, set the Navigator Service public.session parameter to a value of true. Refer to "Portal Configuration Services" for information on how to configure this parameter.

**Parent topic:** Setting up multiple clusters on HP-UX

**Previous topic:** Installing multiple clusters in a single cell on HP-UX
Sharing database domains between clusters on HP-UX

To help provide redundancy and failover support in production environments composed of multiple clusters in a single cell and multiple clusters in different cells, you can share database domains between those clusters. IBM® WebSphere® Portal data is organized into different database domains, with different availability requirements depending on how the production environment is set up. When multiple lines of production are involved and each line of production is implemented as a cluster of servers, sharing database domains ensures that the data is automatically synchronized across the lines of production.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on HP-UX
  - Preparing your HP-UX operating system
  - Preparing the primary node on HP-UX
  - Choosing the type of cluster to create on HP-UX
  - Tune your servers

**Important:** JCR domains and release domains cannot be shared between different Web Content Management clusters or servers. Each distinct cluster or server in your Web Content Management system must use a separate JCR or release domain. For example:

<table>
<thead>
<tr>
<th>Development Server</th>
<th>Authoring Cluster</th>
<th>Staging Server</th>
<th>Delivery Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCR Domain 1</td>
<td>JCR Domain 2</td>
<td>JCR Domain 3</td>
<td>JCR Domain 4</td>
</tr>
<tr>
<td>Release Domain 1</td>
<td>Release Domain 2</td>
<td>Release Domain 3</td>
<td>Release Domain 4</td>
</tr>
</tbody>
</table>

Perform the following steps to share database domains when setting up an environment with multiple clusters:

1. Set up the first cluster (referred to as Cluster A in these instructions).
2. Determine which database domains you want to share with any other clusters in the environment.
3. Install the primary node of the next cluster (Cluster B), and perform the following steps to configure the node to use the shared database domains.
   A. Perform a partial database transfer of the database domains that you are not sharing. For example, if you are sharing only the Customization and Community domains, you would transfer the remaining domains to the database you are using for Cluster B.
   B. Reconfigure the shared database domains on the node to connect to the shared database domains you are using for Cluster A. For example, to connect to the Customization and Community domains for Cluster A, you would connect to those domains from Cluster B.
4. Continue setting up the primary node as described in the cluster instructions.
5. Install the secondary node of Cluster B, and perform the following steps to configure the node to use the shared database domains.
   A. For those database domains that you are not sharing between clusters, reconfigure the domains to connect to the database domains you are using for Cluster B. As in the example for the primary node, if you are sharing only the Customization and Community domains, reconfigure the remaining domains on the secondary node to use the domains of the primary node for Cluster B.
B. Reconfigure the shared database domains on the node to connect to the shared database domains you are using for Cluster A. For example, to connect to the Customization and Community domains for Cluster A, you would connect to those domains from Cluster B.

6. Continue setting up the secondary node as described in the cluster instructions.

**Parent topic:** Setting up a cluster on HP-UX  
**Previous topic:** Setting up multiple clusters on HP-UX

**Related tasks**

Connecting to existing database domains
Setting up a cluster on IBM i5/OS

In a production environment, you can install and configure IBM® WebSphere® Portal on a single (primary) node and then add additional nodes in the future as your company's needs grow. You can also deploy a high availability production environment by creating and configuring clustered deployments of multiple servers running WebSphere Portal managed by IBM WebSphere Application Server Network Deployment.

**Prerequisites**

- [Technotes for installation and configuration issues](#)

Perform the following tasks to set up your production environment on i5/OS:

1. **Preparing prerequisite and corequisite software on IBM i5/OS**
   
   Before creating your high availability production environment, you must prepare your database, your user registry, your deployment manager, and your remote Web server.

2. **Preparing your IBM i5/OS**
   
   View information on setting up your operating system for IBM WebSphere Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

3. **Preparing the primary node on IBM i5/OS**
   
   Before creating your high availability environment, you must install IBM WebSphere Portal on your primary node and then configure your database and your network deployment manager.

4. **Creating the cluster on i5/OS**
   
   After installing IBM WebSphere Portal on the primary node, configuring a remote database, and preparing the primary node to communicate with the Deployment Manager, you can create your static cluster to handle failover requests.

5. **Preparing the Web server when portal is installed on IBM i5/OS**
   
   Install and configure the Web server plug-in, provided by IBM WebSphere Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

6. **Preparing user registries on IBM i5/OS**
   
   Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

7. **Configuring WebSphere Portal to use a user registry on IBM i5/OS in a clustered environment**
   
   Configure user registry security on IBM WebSphere Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

8. **Preparing additional nodes on IBM i5/OS**
   
   After installing and configuring your primary node, you can create your secondary nodes. You must install IBM WebSphere Portal on each node and then configure the node to access the database and user registry before adding it to the cluster.

9. **Optional: Rendering documents on IBM i5/OS**
   
   In order to enable document preview functionality for IBM Lotus Web Content Management and the Common Mail portlet, you must set up an HTML rendering server to work with WebSphere Portal. Because IBM i5/OS does not contain native graphics support, you must install additional fonts to perform the document conversion required by these functions. Document conversion enables WebSphere Portal to convert documents produced by commonly used office programs into Web pages, so that they can be viewed and searched by users online. The additional fonts include an
10. **Tune your servers**

Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

11. **Configuring search in a cluster on i5/OS**

IBM WebSphere Portal provides two distinct search capabilities. You can utilize both types of search capabilities in a clustered environment.

12. **Setting up multiple clusters on IBM i5/OS**

The majority of the steps to build another cluster are the same as when building the first cluster in the same cell, with a few exceptions. Basically, the new profile will be designated as the primary profile, using IBM WebSphere Portal clustering terminology, and will then be used as the basis for the new cluster definition. This duplicates the build process of the first cluster in the cell. During the federation process, if any applications on this new node already exist in the cell (because they are in use by the first cluster), then Deployment Manager will not allow them to be added. After federation, the applications that already exist in the cell are not mapped to the WebSphere.Portal server on the newly federated node, and thus the existing applications must be remapped to this newly federated server to restore its application list. Therefore, depending on the configuration of the new profile, there will likely be some combination of applications shared with the other existing cluster, and some applications unique to this new profile.

13. **Sharing database domains between clusters on IBM i**

To help provide redundancy and failover support in production environments composed of multiple clusters in a single cell and multiple clusters in different cells, you can share database domains between those clusters. IBM WebSphere Portal data is organized into different database domains, with different availability requirements depending on how the production environment is set up. When multiple lines of production are involved and each line of production is implemented as a cluster of servers, sharing database domains ensures that the data is automatically synchronized across the lines of production.

**Parent topic:** Setting up a cluster
Preparing prerequisite and corequisite software on IBM i5/OS

Before creating your high availability production environment, you must prepare your database, your user registry, your deployment manager, and your remote Web server.

Perform the following tasks to prepare prerequisite and corequisite software:

1. **Preparing the WebSphere Application Server Deployment Manager on i5/OS**
   IBM WebSphere Portal provides a customized installation package (CIP) that includes IBM WebSphere Application Server Network Deployment and all required maintenance packages. Use the supplied discs to install WebSphere Application Server Network Deployment on a dedicated system.

2. **Preparing the Web server when portal is installed on IBM i5/OS**
   Install and configure the Web server plug-in, provided by IBM WebSphere Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

Parent topic: Setting up a cluster on IBM i5/OS

Next topic: Preparing your IBM i5/OS
Preparing the WebSphere Application Server Deployment Manager on i5/OS

IBM® WebSphere® Portal provides a customized installation package (CIP) that includes IBM WebSphere Application Server Network Deployment and all required maintenance packages. Use the supplied discs to install WebSphere Application Server Network Deployment on a dedicated system.

Perform the following steps to prepare the WebSphere Application Server Deployment Manager:

1. Install WebSphere Application Server Network Deployment or use the CIP to upgrade an existing installation. Run the following command: cd_root/ISERIES/architecture/ifpackage/NAS/install, where cd_root is the root directory of the disc and architecture is the system's processor architecture.

2. Use the manageprofile commands to create a deployment manager profile; see Profile concepts.

3. Run the following command to start the deployment manager: startManager, from the dmgr_profile_root/bin directory.

4. Use the following URL to launch the network deployment administrative console: http://dmgr_hostname:9060/ibm/console, where dmgr_hostname is the fully qualified host name for the WebSphere Application Server Network Deployment.

5. Log into the deployment manager administrative console.

6. Increase the HTTP connection timeouts for the deployment manager.
   A. Click System Administration > Deployment Manager > Web container transport chains.
   B. Increase the timeout values. For the WCInboundAdmin and WCInboundAdminSecure entries listed in the web container transport chains section, complete the following steps to increase the timeout values:
      1. Click HTTP Inbound Channel.
      2. Change the Read timeout value to 180.
      3. Change the Write timeout value to 180.
      4. Save the configuration changes.

7. Change the timeout request period for the Java Management Extensions (JMX) connector.
   A. Click System administration > Deployment Manager > Administration Services > JMX connectors > SOAPConnector > Custom Properties.
   B. Select the requestTimeout property, and increase the value from 600 to 6000.
   C. Save the configuration changes.

8. Update the maximum Java heap size used by the deployment manager:
   B. Update the value in the Maximum Heap Size field. For information about appropriate heap sizes see the documentation for your operating system in the Performance Guides located on the WebSphere Portal and Web Content Management Product Documentation page. **Note:** If using a 32-bit operating system, you will need to set the heap size to a lower size than a 64-bit operating system.
   C. Click OK and then save your changes.

9. Click Security > Global security and select Enable Application Security. Then save the configuration changes.  
   **Note:** If security is not enabled on your deployment manager, see Enabling security for information before performing this step.

10. Verify that the WebSphere Portal administrative users and administrative group exist in the Deployment Manager cell's user registry. Perform the following steps if you need to create the administrative users and group:
A. Click Users and Groups > Manage Users.
B. Click Create.
C. Type the information for the WebSphere Portal administrative users; for example wpsadmin and wpsbind, and then click Create.
D. Click Users and Groups > Manage Groups.
E. Click Create.
F. Type wpsadmins as the name of the WebSphere Portal administrative group and then click Create.
G. Click the group you just created; for example wpsadmins.
H. Click the Members tab.
I. Click Add Users.
J. Search for the users.
K. Select the users you want to add to the group.
L. Click Add to add the users to the group.
M. Click Close when you are done adding users to the group.
N. Log out of the administrative console.


12. Run the following tasks to stop and restart the deployment manager:
   A. stopManager -username admin_userid -password admin_password, from the dmgr_profile_root\bin directory
   B. startManager, from the dmgr_profile_root\bin directory

Parent topic: Preparing prerequisite and corequisite software on IBM i5/OS
Next topic: Preparing the Web server when portal is installed on IBM i5/OS

Related tasks
Back up the Deployment Manager profile on i5/OS
Restoring the Deployment Manager profile
Preparing the Web server when portal is installed on IBM i5/OS

Install and configure the Web server plug-in, provided by IBM® WebSphere® Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

- **Prerequisites**
  - Preparing the WebSphere Application Server Deployment Manager on i5/OS
  - Preparing prerequisite and corequisite software on IBM i5/OS
  - Preparing your IBM i5/OS
  - Preparing the primary node on IBM i5/OS

Perform the following steps to install and configure your Web server:

1. Install and configure the Web server; refer to the Web server documentation for information.
2. If using IBM Lotus® Domino®, edit the **NOTES.INI** file on the Web server. Set the `HTTPEnableConnectorHeaders` parameter to 1.
3. If using IBM HTTP Server or Apache Server, edit the **httpd.conf** file on the Web server. Set the `AllowEncodedSlashes` directive to `On`; the directive should be added at the root level as a global directive.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>See the appropriate HTTP Server documentation</td>
<td>IBM HTTP Server</td>
</tr>
<tr>
<td>See the appropriate Apache Server documentation</td>
<td><code>AllowEncodedSlashes</code> directives</td>
</tr>
</tbody>
</table>

4. Stop the Web server.
5. Install and configure the Web server plug-in on the system where the Web server is located using the Plug-ins installation wizard provided with WebSphere Application Server. Refer to Selecting a Web server topology diagram and roadmap for information. **If using WebDAV**: After successfully installing the Web server plug-in, locate and open your **plugin-cfg.xml** file and set `AcceptAllContent` to `true`.

**Important**: Depending on how you use the Web server, you may need to adjust the `ServerIOTimeout` value, which defines how long the plug-in should wait for a response from the application. The recommended minimum value is 60 but you may need to adjust this value higher if you are retrieving data from a database. To update this value, locate and open your **plugin-cfg.xml** file and set `ServerIOTimeout` to a value that is appropriate for your business needs. For additional information, see Common questions about the Web server plug-in.

6. If you are using Sun Web Server Version 7 update 8, read Technote 1448262 and perform the steps to resolve the HTTP 408/409 error.
7. Start the Web server.
Preparing your IBM i5/OS

View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on IBM i5/OS

WebSphere Portal can be installed locally or remotely using a Windows workstation.

You need the following information to install remotely:
- A supported version of Microsoft® Windows®
- Workstation CD-ROM drive
- TCP/IP connection to the IBM i5/OS system where you will install WebSphere Portal
- The IBM i5/OS server must be in an unrestricted state
- A valid user ID and password on the IBM i5/OS system
- A user profile with a user type (user class) of *SECOFR (other than QSECOFR) to install and configure WebSphere Portal

You need the following information to install locally:
- IBM i5/OS CD-ROM drive
- The IBM i5/OS server must be in an unrestricted state
- A valid IBM i5/OS user ID and password
- A user profile with a user type (user class) of *SECOFR (other than QSECOFR) to install and configure WebSphere Portal

**Parent topic:** Setting up a cluster on IBM i5/OS

**Previous topic:** Preparing prerequisite and corequisite software on IBM i5/OS

**Next topic:** Preparing the primary node on IBM i5/OS
Preparing the primary node on IBM i5/OS

Before creating your high availability environment, you must install IBM® WebSphere® Portal on your primary node and then configure your database and your network deployment manager.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on IBM i5/OS
  - Preparing your IBM i5/OS

Perform the following tasks to prepare your primary node:

1. **Installing WebSphere Portal on IBM i5/OS on the primary node**
   IBM WebSphere Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

2. **Configure WebSphere Portal to use a remote database**
   For high availability, using a remote database is preferred. For improved performance databases can be distributed across multiple database servers. Databases should also be configured for failover. Configuring the database for failover is beyond the scope of this documentation. Refer to the database product documentation for the most authoritative guidance about setting up databases for failover.

3. **Configuring the primary node to communicate with the deployment manager on i5/OS**
   After installing IBM WebSphere Application Server Network Deployment and installing IBM WebSphere Portal on the primary node, you must configure the primary node and the deployment manager to communicate with each other.

4. **Removing search collections on IBM i5/OS**
   If you plan to use search in a cluster, you must configure a remote search server; see Configuring Portal Search in a cluster for information. If you created any search collections, you will need to recreate them on the remote search server. If your search collection has data, export the collection before deleting it and then import the data on the remote server.

**Parent topic:** Setting up a cluster on IBM i5/OS  
**Previous topic:** Preparing your IBM i5/OS  
**Next topic:** Creating the cluster on i5/OS
Installing WebSphere Portal on IBM i5/OS on the primary node

IBM® WebSphere® Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

Review the Supported hardware and software requirements for your operating system and ensure that all hardware and software matches the minimum product level before installing WebSphere Portal. Review Installation methods, options, and sources before installing to determine your installation source location and installation method. Perform the following steps to install WebSphere Portal and create a custom profile for the node that will be added to the cluster. If you already installed WebSphere Portal, you can skip to the last step to create the custom profile.

1. Type `ping yourserver.yourcompany.com` on a command line to verify that your fully qualified host name is properly configured.
2. Type `ping localhost` on a command line to verify that your network is properly configured.
3. Setup a static IP address on the server where you will install WebSphere Portal.
4. If you are installing with an existing WebSphere Application Server instance, ensure it is installed at the supported level and has the following features installed:
   - **Restriction:** Installing WebSphere Portal Version 6.1.0.1 or higher on an existing WebSphere Application Server Version 7.0 instance where WebSphere Portal Version 7.0 was installed is unsupported.
   - **Important:** Besides ensuring that the existing WebSphere Application Server instance is at the supported level, you will also need to ensure that Apache Derby is installed at the supported level before installing WebSphere Portal. See WebSphere Portal detailed system requirements for supported levels.

   - Application Server
   - Administration
   - Scripted Administration
   - Administrative Console
   - Ant and Deployment Tools
   - Deploy Tool
   - Ant Utilities

5. Choose one of the following installation commands based on the installation method you want to use to install WebSphere Portal; see Installation Methods for more information:
   - **Advance installation parameters:** To customize your installation, you can add various parameters to your installation commands; for example, you can specify your port information. See Advanced installation parameters for information.
   - **Restriction:** When defining your installation path, the ONLY supported characters are ASCII letters [A-Z and a-z], numbers [0-9], slashes [/ or \, depending on your operating system], and the dash [-].

   Table 1. Installation task options

<table>
<thead>
<tr>
<th>Installation type</th>
<th>Task</th>
</tr>
</thead>
</table>

1501
**Graphical user interface**

<table>
<thead>
<tr>
<th>Command</th>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>install400.bat</td>
<td>Optional attribute: WebSphere Application Server profiles and configurations are performed with the Classic 64-bit JVM by default. To install and configure the portal profile with J9 32-bit JVM, add the <code>-W enableClassicJVM.active=false</code> attribute to your installation command.</td>
<td></td>
</tr>
</tbody>
</table>

**Console mode remote**

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>install400.bat -console</td>
</tr>
</tbody>
</table>

**Console mode local**

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>install.sh</td>
</tr>
</tbody>
</table>

**Silent install remote**

<table>
<thead>
<tr>
<th>Command</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>install400.bat -options &quot;path_to_file\response_filename&quot;, where <code>path_to_file</code> is the full path to the response file and <code>response_filename</code> is the name of the file. A sample install response file (installresponse.txt) and a sample uninstall response file (uninstallresponse.txt) are located in the setup CD root directory. <strong>Important:</strong> Do not place the response file in a path that contains a space and do not put a space in the file name.</td>
<td></td>
</tr>
</tbody>
</table>

**Silent Install local**

<table>
<thead>
<tr>
<th>Command</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>install.sh -options &quot;path_to_file\response_filename&quot;, where <code>path_to_file</code> is the full path to the response file and <code>response_filename</code> is the name of the file. A sample install response file (installresponse.txt) and a sample uninstall response file (uninstallresponse.txt) are located in the setup CD root directory. <strong>Important:</strong> Do not place the response file in a path that contains a space and do not put a space in the file name.</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If the GUI or console mode installation program fails to detect ports for either WebSphere Application Server or WebSphere Portal, a warning message displays and the installer offers another chance to enter the values. If the silent installation fails to detect ports for either WebSphere Application Server or WebSphere Portal, the installer will exit.

6. Verify your installation was successful; access WebSphere Portal using the `http://yourserver:yourport/wps/portal` format. Run the `ConfigEngine.sh list-server-ports -DWwasPassword=password` task from the `wp_profile_root`/ConfigEngine directory to generate the `wp_profile_root`/ConfigEngine/log/wp_PortMatrix.txt file that lists the WebSphere Portal ports for your installation.

7. Optional: After the WebSphere Portal installation completes successfully, run the `ConfigEngine.sh configure-express -DPortalAdminPwd=password -DWwasPassword=password` task from the `wp_profile_root`/ConfigEngine directory, if you want to enable the sample IBM Lotus Web Content Management content, such as internet and intranet sample sites.

**Restriction:** Run the `configure-express` task before configuring your database, user registry, context root, security, etc. If you ran any tasks other than the install task, do not run this task.

**Note:** See [http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_ovw.html](http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart.ovw.html) for tutorials on how to use the sample content.

The sample content includes:
- Creates a group called contentAuthors; members of this group are given privileges to create content in the sample
Internet and intranet sites. Navigate to the Administration area and then click Access > Users and Groups.

- Creates two new Web Content Management Libraries: "Internet Web Content 6.1.0" and "Intranet Web Content 6.1.0".

Navigate to the Administration area and then click Portal Content > Web Content Libraries.

- Adds a portlet filter and applies the filter to various portlets in the sample Internet and intranet sites. You can see the definition of the filter in the WebSphere Application Server Administration console and examining the custom resources under the Environment area.

- Creates two new theme policies: InternetStyle and IntranetStyle. These styles are applied to sample Internet and intranet sites. Navigate to Theme Customizer and then select the style.

- Creates several portlet clones of the Web Content Management rendering portlet. These portlet clones are used on sample Internet and intranet sites.

- Creates two virtual portals with context roots of wps/portal/intranet and wps/portal/internet. These are the sample Internet and intranet sites. Go to http://yourserver:yourport/wps/portal/internet and http://yourserver:yourport/wps/portal/intranet to access them.

- Creates several sample credential slots, including "Default slot for E-mail", "Default slot for Feeds", "Default slot for Miscellaneous", "Default slot for Web Clipping", and "Default slot for Web Content Management". Navigate to the Administration area and then click Access > Credential Vault > Manage System Vault Slots.

- Modifies the portlet palette to include some new portlets. Click the Portlets link to see the portlet palette.

Optional: If you ran the configure-express task, the owner of the items in the Web content libraries containing the Internet and Intranet Site Template content will be listed as uid=xyzadmin,o=defaultWIMFileBasedRealm. To update the owner information for these items to correspond to your portal administrator ID, complete the following steps:

A. Edit the wp_profile_root/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties file.

B. Add the following line to the file:

```
uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
```

Replace portal_admin_DN with the distinguished name of the portal administrator.

C. Save your changes and close the file.

D. Run the ConfigEngine.sh action-express-memberfixer -DPortalAdminPwd=password -DWasPassword=password task, located in the wp_profile_root/ConfigEngine directory.

If you installed WebSphere Portal on an i5/OS Version 6.1 system, install the MF45016 fix; see Fix Central to access the fix.

Set the following two cache properties under the WP_CacheManagerService resource environment provider in the WebSphere Application Server Administrative Console; see Setting service configuration properties for information:

- cacheglobal.softref=true
- cacheglobal.lifetime=1200

Parent topic: Preparing the primary node on IBM i5/OS

Next topic: Configure WebSphere Portal to use a remote database

Related concepts

- Installation methods
- IBM Support Assistant Lite for WebSphere Portal

Related reference

- Advanced installation parameters
Configure WebSphere Portal to use a remote database

For high availability, using a remote database is preferred. For improved performance databases can be distributed across multiple database servers. Databases should also be configured for failover. Configuring the database for failover is beyond the scope of this documentation. Refer to the database product documentation for the most authoritative guidance about setting up databases for fail over.

**Password considerations when transferring data manually**

For security reasons, you should not store passwords in the wkplc.properties and wkplc_comp.properties. Edit each of the properties files prior to running a configuration task, inserting the passwords needed for that task. After the task has run, delete all passwords from each file. For more information, see Deleting passwords from configuration scripts.

Alternatively, you can specify the password on the command line using the following syntax:

- **Windows**: `ConfigEngine.bat task_name -Dpassword_property_key=password_value`
- **UNIX**: `./ConfigEngine.sh task_name -Dpassword_property_key=password_value`
- **i5/OS**: `ConfigEngine.sh task_name -Dpassword_property_key=password_value`

As with other properties, each password property must have the `-D` prefix and be set equal to (=) a value. If you have multiple properties in a single command, use a space character between each `-Dproperty=value` setting.

**Prerequisites**

- Installing WebSphere Portal on IBM i5/OS on the primary node
- [Technote for database connectivity issues](#)

**Preparing DB2 for i5/OS**

View information on installing and setting up DB2 for i5/OS to work with WebSphere Portal.

**Parent topic:** Preparing the primary node on IBM i5/OS

**Previous topic:** Installing WebSphere Portal on IBM i5/OS on the primary node

**Next topic:** Configuring the primary node to communicate with the deployment manager on i5/OS

**Related information**

Troubleshooting WebSphere Portal Version 6.1 databases
Preparing DB2 for i5/OS

View information on installing and setting up DB2 for i5/OS to work with WebSphere Portal.

1. Preparing for creation of databases
   View the mandatory preparation tasks prior to creating databases that are required by WebSphere Portal.

2. Create user profiles
   View information on setting up user profiles for DB2 for i5/OS to work with WebSphere Portal.

3. Creating remote databases
   A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

4. Modifying database properties
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. Transferring DB2 for i5/OS manually
   View the steps to manually transfer data to the IBM® DB2 Universal Database™ for iSeries® database you have set up. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

6. Verifying database connections
   After WebSphere Portal is configured to work with your database, test the database connection to ensure that it operates correctly.

**Parent topic:** Configure WebSphere Portal to use a remote database
Preparing for creation of databases

View the mandatory preparation tasks prior to creating databases that are required by WebSphere Portal. The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Preparing database domain property values prior to creating database names and schemas. Before you manually configure your database schemas, you must first edit your database domain properties files.

1. **Preparing database domain property values prior to creating database names and schemas.** Before you manually configure your database schemas, you must first edit your database domain properties files.

   A. Locate the following files and create a backup copy of each before changing any values:
   - `wp_profile_root/ConfigEngine/properties/wkplc.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_comp.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties`

   B. The WebSphere Portal database can be used to hold information for applications such as Personalization (Feedback) and LikeMinds. To prepare the database to hold such application information, you should use similar naming conventions for property values such as `release.DbName`. Here are some examples:

   - `release.DbName=hostname/ WP61REL`
   - `community.DbName=hostname/ WP61COM`
   - `customization.DbName=hostname/ WP61CUS`
   - `jcr.DbName=hostname/ WP61JCR`
   - `feedback.DbName=hostname/ WP61FBK`
   - `likeminds.DbName=hostname/ WP61LKM`

   When you create a schema, you must use the following schema naming conventions on the i5/OS system: **Note:** The default schema names may be used with the product.

   - Length cannot exceed 10 characters
   - All alphanumerical characters are allowed ("A" through "Z" and "1" through "0")
   - The following characters are invalid:
     - spaces
     - null values
     - asterisk (*)
     - quotation marks ("")
     - colon (:)
     - greater than symbol (>)
     - less than symbol (<)
     - vertical bar (!)
- plus sign (+)
- semicolon (;)
- single quotation mark (’)
- question mark (?)

Notes:
- Make sure you know what valid schema names are and do not use a schema name which already exists on the local or remote system. Follow the documentation of the target database management system in order to define a valid schema name as restrictions apply. Note that the Create WebSphere Portal wizard will automatically check schema names for you.
- For more information on database and schema naming conventions, refer to DB2 Universal Database for System i5 topic in the System i5 information center.

- Do not change any settings other than those that are specified in these substeps.
- If you are using a remote database, be sure to enter the values for the remote server.
- Use / instead of \ for i5/OS.
- Some values, shown here in italics, might need to be modified to your specific environment.
- Password considerations: For security reasons, you should not store passwords in the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files. It is recommended that you edit each of the properties files prior to running a configuration task, inserting the passwords needed for that task. Then, after the task has run, you should delete all passwords from each file.
- There might be additional database properties other than those listed here. Only change the properties within this table; skip all other properties.
- Depending on which database domain has to be configured, the variable dbdomain may need to be replaced by:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds

C. Use a text editor to open the properties files and enter the values that are appropriate for your environment. You can also modify each properties file locally on your System i5 system by typing the following on an OS/400 command line in a 5250 session: EDTF 'wp_profile_root/ConfigEngine/properties/property_filename.properties'

Note: You must have a user profile on the System i5 server and must have at least *USE special authority to edit the properties file.

Tip: The steps for transferring data to another supported database section provide instructions for manually transferring data. Instead of performing the following steps, you can use the configuration wizard, which is a graphical user interface, to transfer data to another supported database.

The following properties must be changed before creating a database name and schema on a local or remote System i5 server.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.
   A. For dbdomain.DbType, type db2_iseries.
   B. For dbdomain.DbName, type the name of the WebSphere Portal domain database. Note: This value is also the database element in the dbdomain.DbUrl property.
   C. For dbdomain.DbSchema, type the schema name of the database domain. Note: Review your target database management system documentation to define a valid schema name. Some database management systems have
schema name restrictions that you need to understand.

D. For \texttt{dbdomain.DataSourceName}, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
- releaseDS
- communityDS
- customizationDS
- jcrDS
- lmdbDS
- feedback

E. For \texttt{dbdomain.DbUrl}, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. \textbf{Note:} The database element of this value should match the value of DbName.

F. For \texttt{dbdomain.DbUser}, type the user ID for the database administrator.

G. For \texttt{dbdomain.DbPassword}, type the password for the database administrator.

H. For \texttt{dbdomain.DBA.DbUser}, type the database administrator user ID for privileged access operations during creation of the database.

I. For \texttt{dbdomain.DBA.DbPassword}, type the database administrator password for privileged access operations during creation of the database.

3. Save and close the file.

4. Update the following properties in the file \texttt{wkplcDbType.properties}.
   A. For \texttt{db2_iseries.DbDriver}, type the name of the class that \texttt{SqlProcessor} uses to import SQL files.
   B. For \texttt{db2_iseries.DbLibrary}, type the directory and name of the .zip or .jar file that contains the JDBC driver class.
   C. For \texttt{db2_iseries.JdbcProviderName}, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.
   D. For \texttt{db2_iseries.DbDriverType}, type the number representing the driver type for the database.

5. Save and close the file.

6. Update the following property in the file \texttt{wkplc.properties}.
   A. For \texttt{WasPassword}, type the password for the WebSphere Application Server security authentication used in your environment.

7. Save and close the file.

\textbf{Parent topic:} Preparing DB2 for i5/OS  
\textbf{Next topic:} Create user profiles
Create user profiles

View information on setting up user profiles for DB2 for i5/OS to work with WebSphere Portal.

Before you begin:
- The user profile for the database owner should be different from the administrator user profile used to perform the installation. The administrator user profile may have more authority than is required and usually belongs to an individual, whereas the database user profile may have minimal authority and can be shared.
- Create a database user profile that does not require a password change over a period of time. If the password for the database user profile changes, WebSphere Portal must be re-configured to use the new password.
- Create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

- Prerequisites
  - Preparing for creation of databases

To create user profiles, follow the instructions that are provided with the DB2 for i5/OS documentation.

Parent topic: Preparing DB2 for i5/OS
Previous topic: Preparing for creation of databases
Next topic: Creating remote databases
Creating remote databases

A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

Before you begin:
- The user ID and password used must have the authority to create database libraries on the remote System i5 machine.
- For every property instance of a database that uses "LOCAL/schema, replace it with HostName/schema. For example, the default database and database library name for WebSphere Portal release domain is release.DbName=wpsdb. If you wanted to create this database library on a remote database, change the default value to release.DbName=hostname/wpsdb.

- **Prerequisites**
  - Preparing for creation of databases
  - Create user profiles

To create all the domain database libraries, perform the following steps:

1. Start a 5250 session on the remote database machine.
2. Type the i5/OS command **WRKRDBDIRE** to display the Relational Database Directory Entry for Remote Location "LOCAL" and make a note of the value displayed.
3. Sign off from the 5250 session.
4. Start a 5250 session on the local machine where WebSphere Portal is installed.
5. Create a Relational Database Directory Entry on the local system for the remote system using i5/OS command **WRKRDBDIRE**.
6. Add an entry with the following values:
   - **Relational database**
     - The remote relational database. Use the value noted from the prior step.
   - **Relational database alias**
     - The hostname. Use the short TCP/IP hostname of the remote system
   - **Remote location**
     - The domain qualified hostname. Use the full TCP/IP hostname of the remote system
   - **Type**
     - IP
   - **Port number or service name**
     - DRDA
   - **Remote authentication method**
     - Preferred method: ENCRYPTED
     - Allow lower authentication: ALWLOWER
7. Create the required DB2 packages on the remote database machine by running the following command from the local machine:
   ```
   JAVA CLASS(com.ibm.db2.jdbc.app.DB2PackageCreator) PARM('rdb_alias' 'userid' 'password')
   PROP((jdbc.drivers 'com.ibm.as400.access.AS400JDBCDriver'))
   ```
   where **rdb_alias** matches the name of the Relational Database Entry you created in step 2, where **userid** is the database administrator user ID on the remote machine, and where **password** is the database administrator password on the remote machine. The output should be:
   
   Java program completed
8. Press F3 to exit Java Shell Display.
9. Sign off from the 5250 session.
10. Start a 5250 session on the remote database machine.
11. Verify the required DB2 packages were created by running the command `WRKOBJ OBJ(QGPL/QSQCL*) OBJTYPE(*SQLPKG)`

```
<table>
<thead>
<tr>
<th>Object</th>
<th>Type</th>
<th>Library</th>
<th>Attribute</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>QSQCLIPKGA</td>
<td>*SQLPKG</td>
<td>QGPL</td>
<td>PACKAGE</td>
<td></td>
</tr>
<tr>
<td>QSQCLIPKGC</td>
<td>*SQLPKG</td>
<td>QGPL</td>
<td>PACKAGE</td>
<td></td>
</tr>
<tr>
<td>QSQCLIPKGL</td>
<td>*SQLPKG</td>
<td>QGPL</td>
<td>PACKAGE</td>
<td></td>
</tr>
<tr>
<td>QSQCLIPKGN</td>
<td>*SQLPKG</td>
<td>QGPL</td>
<td>PACKAGE</td>
<td></td>
</tr>
<tr>
<td>QSQCLIPKGS</td>
<td>*SQLPKG</td>
<td>QGPL</td>
<td>PACKAGE</td>
<td></td>
</tr>
</tbody>
</table>
```

12. Start a 5250 session on the local machine where WebSphere Portal is installed.
13. On the command line, enter the following to change directories: `cd wp_profile_root/ConfigEngine`.
14. Press Enter.
15. Change the property values in the configuration properties files before entering the following on the command line:

```
ConfigEngine.sh create-database
```
16. Press Enter.

**Parent topic:** Preparing DB2 for i5/OS  
**Previous topic:** Create user profiles  
**Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- Prerequisites
  - Preparing for creation of databases
  - Create user profiles
  - Creating remote databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

Tip: For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Working with properties files:
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:

  - release.DbName=release
  - jcr.DbName=jcrdb
  - feedback.DbName=fdbkdb
  - likeminds.DbName=lmdb
  - community.DbName=commdb
  - customization.DbName=custdb

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
  - dbdomain.DbType
- **dbdomain**.DbName
- **dbdomain**.DbUrl
- **dbdomain**.DbSchema

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If **DbUser**, **DbUrl**, and **DbPassword** are not the same across domains, the value for **DataSourceName** must differ from the **DataSourceName** of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   - `wp_profile_root/ConfigEngine/properties/wkplc.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_comp.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties`

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In **wkplc_comp.properties**, most properties are repeated for each domain.

2. Use a text editor to open the properties file `wkplc_comp.properties` and modify the values to correspond to your environment.
   A. For **dbdomain**.DbType, type `db2_iseries`.
   B. For **dbdomain**.DbName, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the **dbdomain**.DbUrl property.
   C. For **dbdomain**.DbSchema, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   D. For **dbdomain**.DataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For **dbdomain**.DbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:** The database element of this value should match the value of DbName.
   F. For **dbdomain**.DbUser, type the user ID for the database administrator.
   G. For **dbdomain**.DbPassword, type the password for the database administrator.

3. **(Optional)** When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in **wkplc_comp.properties**. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.
   A. For **source.domain**.DbType, type of the database you are currently configured to use. The value for **source.domain**.DbType is Derby by default.
   B. For **source.domain**.DbName, type the name of the database domain you are currently using.
   C. For **source.domain**.DbSchema, type current schema identifier for objects within the database for this domain.
   D. For **source.domain**.DataSourceName, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.
E. For \textit{source.domain}.DbUrl, type the url currently used to access your database.

F. For \textit{source.domain}.DbUser, type the name of the user accessing this database.

G. For \textit{source.domain}.DbPassword, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file \texttt{wkplc_dbtype.properties}.
   
   A. For db2\_iseries.DbDriver, type the name of the class that \texttt{SqlProcessor} uses to import SQL files.

   B. For db2\_iseries.DbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For db2\_iseries.JdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file \texttt{wkplc.properties}.
   
   A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

\textbf{Parent topic:} Preparing DB2 for i5/OS  
\textbf{Previous topic:} Creating remote databases  
\textbf{Next topic:} Transferring DB2 for i5/OS manually
Transferring DB2 for i5/OS manually

View the steps to manually transfer data to the IBM® DB2 Universal Database™ for iSeries® database you have set up. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.

- **Prerequisites**
  - Preparing for creation of databases
  - Create user profiles
  - Creating remote databases
  - Modifying database properties

**Steps for transferring data to another supported database**

1. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td>stopServer server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td>stopServer WebSphere_Portal -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

2. Validate configuration properties using the following commands:

   ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password

   ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password

3. Transfer the database: **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   **A.** Enter the following command:

   ```sh
   ConfigEngine.sh database-transfer -DWasPassword=password
   ```

   **Note:**

   - To select specific database domains to transfer, modify the -DTransferDomainList specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:

   ```sh
   ./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
   ```

   - Use SBMJOB to submit the Qshell script as a batch job to run in *BASE pool when *INTERACT pool does not have 1GB or more of allocated memory. For example:

   ```sh
   SBMJOB CMD(STRQSH CMD(ConfigEngine.sh database-transfer -DWasPassword=password))
   ```

   **B.** After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files and then repeat this step.
4. Enter the following command to start the WebSphere Portal server:

```
startServer WebSphere_Portal
```

If you have performed database transfer in a clustered environment, you must explicitly copy the `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm/icm.properties` file from the primary node on which the database-transfer task was processed to all secondary nodes. This ensures that the secondary nodes have the new JCR database values specified in the `icm.properties` file.

1. Stop the portal server on the secondary node.
2. Copy the `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm/icm.properties` file from the primary node and replace the `icm.properties` file on the secondary node with the new file from the primary node.
3. Start the portal server on the secondary node.

**Parent topic:** Preparing DB2 for i5/OS

**Previous topic:** Modifying database properties

**Next topic:** Verifying database connections
Verifying database connections

After WebSphere Portal is configured to work with your database, test the database connection to ensure that it operates correctly.

- **Prerequisites**
  - Preparing for creation of databases
  - Create user profiles
  - Creating remote databases
  - Modifying database properties
  - Transferring DB2 for i5/OS manually

You can verify the connection from a browser or from a command line. To verify that WebSphere Portal is running from a browser, open the portal in a Web browser: http://hostname.yourco.com:port_number/wps/portal, where hostname.yourco.com is the fully qualified host name of the machine where WebSphere Portal is running and port_number is the transport port that is created by IBM® WebSphere® Application Server.

There may be an error if any of the following conditions appear.

- When trying to access the portal you get a 503 error.
- If you had any locale problems with your database, you could see invalid characters, such as ????, after logging in. This may happen if the character set of the database is not UTF-8 compliant.
- If something went wrong with the data that was transferred, you may not be able to login. WebSphere Portal will indicate you entered an invalid user ID and password even though you know it is valid.

Verify the connection from a command line by completing the following steps:

1. Start a 5250 session on the local machine where WebSphere Portal is installed.
2. For WebSphere Portal on WebSphere Application Server (UserData path), enter the following on the command line: `cd wp_profile_root/ConfigEngine`
3. Enter the following command: `ConfigEngine.sh validate-database-connection -DTransferDomainList=release,community,customization,jcr,feedback,likeminds -DWasPassword=password`

For security reasons, you should not leave passwords in the wkplc_comp.properties file. Edit the file prior to running a configuration task and insert the passwords that are needed for that task. After the task has run, delete all passwords from the file.

Alternatively, you can specify the password on the command line rather than update the wkplc_comp.properties file. For example: `ConfigEngine.sh -DPortalAdminPwd=password -DWasPassword=password validate-wps-admin-login`

When installing WebSphere Portal, the passwords in the wkplc_comp.properties file are automatically removed after configuration.

**Parent topic:** Preparing DB2 for i5/OS

**Previous topic:** Transferring DB2 for i5/OS manually
Configuring the primary node to communicate with the deployment manager on i5/OS

After installing IBM® WebSphere® Application Server Network Deployment and installing IBM WebSphere Portal on the primary node, you must configure the primary node and the deployment manager to communicate with each other.

- Prerequisites
  - Installing WebSphere Portal on IBM i5/OS on the primary node
  - Configure WebSphere Portal to use a remote database

Perform the following step to configure WebSphere Portal to communicate with the deployment manager:

Perform the following steps to collect files from the primary node and copy them to the deployment manager:

1. Run the ConfigEngine.sh collect-files-for-dmgr -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory of the primary node, to create an archive or compressed file containing all the files which need to be copied to the Deployment Manager. **Note:** The archive or compressed file will be placed in the wp_profile_root/filesForDmgr directory and the file will be called filesForDmgr.zip.

2. Stop the deployment manager.

3. Expand the filesForDmgr.zip file into the installation root directory of the Deployment Manager; for example this may be a subdirectory under the AppServer directory and it will contain the bin and profileTemplates directories. **Note:** If the Deployment Manager profile was not created in the default AppServer/profiles/Dmgr01 directory, then the metadata_wkplc.xml file, located in the AppServer/profiles/Dmgr01/config/.repository/metadata_wkplc.xml directory in the zip file, must be placed into the correct Deployment Manager profile directory.

4. If the Deployment Manager profile is running on the same application server where WebSphere Portal is installed, remove the com.ibm.ws.portletcontainer.deploytask_6.1.5.jar file from the AppServer/plugins directory.

5. Start the deployment manager.

**Parent topic:** Preparing the primary node on IBM i5/OS

**Previous topic:** Configure WebSphere Portal to use a remote database

**Next topic:** Removing search collections on IBM i5/OS

**Related tasks**

Configuring Portal Search in a cluster on i5/OS
Removing search collections on IBM i5/OS

If you plan to use search in a cluster, you must configure a remote search server; see Configuring Portal Search in a cluster for information. If you created any search collections, you will need to recreate them on the remote search server. If your search collection has data, export the collection before deleting it and then import the data on the remote server.

- Prerequisites
  - Installing WebSphere Portal on IBM i5/OS on the primary node
  - Configure WebSphere Portal to use a remote database
  - Configuring the primary node to communicate with the deployment manager on i5/OS

Perform the following step to remove the search collection:

Perform the following steps to delete all existing search collections from the primary node:

1. Log on to WebSphere Portal.
2. Navigate to Administration > Search Administration > Manage Search and then click Search Collections.
3. Click the Delete Collection icon for each search collection and then click OK until they are all deleted.
4. Restart the WebSphere_Portal server and then navigate back to the Search Collections page to verify that all search collections have been deleted.

Parent topic: Preparing the primary node on IBM i5/OS

Previous topic: Configuring the primary node to communicate with the deployment manager on i5/OS
Creating the cluster on i5/OS

After installing IBM® WebSphere® Portal on the primary node, configuring a remote database, and preparing the primary node to communicate with the Deployment Manager, you can create your static cluster to handle failover requests.

**Prerequisites**

- Preparing prerequisite and corequisite software on IBM i5/OS
- Preparing your IBM i5/OS
- Preparing the primary node on IBM i5/OS

Perform the following steps to create the cluster:

1. If the Deployment Manager is configured to use a stand-alone LDAP user registry, update the wkplc.properties file, located in the `wp_profile_root/ConfigEngine/properties` directory, on the primary node with the stand-alone LDAP user registry property values from the Deployment Manager. You can find these settings under the VMM Stand-alone LDAP configuration heading. **Important:** Ensure that you set `WasUserId` and `WasPassword` to the Deployment Manager user ID and password.

2. Stop the server1 and WebSphere_Portal servers on the primary node and ensure that the following parameters are set correctly in the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory:
   - **Note:** Although you can add these parameters directly to any task that you run while creating your cluster, you may want to add them to the properties file while creating your cluster and then remove them when you are finished to keep your environment secure.
     - A. Set `WasSoapPort` to the port used to connect remotely to the deployment manager.
     - B. Set `WasRemoteHostName` to the full host name of the server used to remotely connect to the deployment manager.
     - C. Verify that `WasPassword` is set to your deployment manager password.
     - D. Verify that `PortalAdminPwd` is set to your WebSphere Portal password.
     - E. Verify that `ClusterName` is set.
     - F. Verify that `PrimaryNode` is set to `true`.

3. Make a backup copy of the `wp_profile_root/config/cells/cell_name/wim/config/wimconfig.xml` and `wp_profile_root/config/cells/cell_name/wim/model/wimxmlextension.xml`, if available, files.

4. Run the `ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=dmgr_password task`, from the `wp_profile_root/ConfigEngine` directory of the primary node. **Note:** If you want to specify custom ports for the nodeagent, add the `-DPortPropsFile=full to portfile` parameter to the `cluster-node-config-pre-federation` task. You can use the ports files that are found on the Setup CD for WebSphere_Portal and server1 as a guide.

   **Note:** You may receive a message about accepting an SSL signer certificate. Failure to accept the SSL signer certificate will cause the script to fail. Alternatively, the `com.ibm.ssl.enableSignerExchangePrompt` flag can be enabled in the `ssl.client.props` file for "DefaultSSLSettings" in order to allow acceptance of the signer during the connection attempt.

   **Warning:** If the `cluster-node-config-pre-federation` fails for any reason, you must perform the following steps before rerunning the task:
   - A. Remove the node if the `AddNode` task succeeded.
   - B. Log on to the deployment manager and perform the following steps if the items exist:
     1. Remove all enterprise applications.
     2. Remove the WebSphere_Portal server definition.
3. Remove the WebSphere Portal JDBC Provider.

**Note:** If you are migrating from a version prior to Version 6.1.0 that contains Lookaside data, read the following technote: Migrating with Lookaside Data.

5. Optional: If you configured a database user registry or a property extension (lookaside) database, run the following task to set up access to the database drivers:

**Note:** You will also need to run this task if you migrated the primary node from a release prior to Version 6.1.0, even if you did not configure a database user registry or a property extension (lookaside) database.

A. Set the property value for `federated.db.DbType` if using a database user registry or set the property value for `la.DbType` if using a property extension database in the `wkplc.properties` file.

**Note:** If you migrated the primary node from a version prior to Version 6.1.0 and you are not using a database user registry or a property extension database, set the property value for `federated.db.DbType` to the same value that is in the `wkplc.properties` file on the primary node.

B. Run the `ConfigEngine.sh wp-node-prep-vmm-db-secured-environment` task from the `wp_profile_root/ConfigEngine` directory to create the variable used to access the VMM database jars. `Note: VmmNodeName` is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The `db_type` in `db_type.NodeDbLibrary` should be set to the type of database you are using, for example `db2`.

- **DB2 for i5/OS Type 2 driver:** `/QIBM/ProdData/Java400/ext/db2_classes.jar`
- **DB2 for i5/OS Type 4 driver:** `/QIBM/ProdData/HTTP/Public/jt400/lib/jt400.jar`

6. Run the `ConfigEngine.sh cluster-node-config-post-federation` task.

7. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell's user registry. Run the `ConfigEngine.sh wp-change-portal-admin-user` task from the `wp_profile_root` directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

**Attention:** The password value for `-DWasPassword` is the Deployment Manager administrative password.

**Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

**Fast path:** If the value for `newAdminGroupId` contains a space; for example `Software Group`, open the `wkplc.properties` file and add the values for `newAdminId`, `newAdminPw`, and `newAdminGroupId`. Save your changes and run the `ConfigEngine.bat wp-change-portal-admin-user` task. If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user` task until after the cluster-node-config-cluster-setup (static cluster) or cluster-node-config-dynamic-cluster-setup (dynamic cluster) task completes. After running the `wp-change-portal-admin-user` task, start or restart the WebSphere_Portal server to use the updated administrator user ID.

**Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the `wp-change-portal-admin-user` task. `-DnewAdminPw` is an optional parameter to update the Administrative password in the `wkplc.properties` file if required.

8. Run the `ConfigEngine.sh cluster-node-config-cluster-setup` task.
9. Configure the cluster to use an external Web server to take advantage of features such as workload management.
   Choose one of the following options:
   - Recommended remote distributed installation
   - Local distributed installation
   - Setting up multiple clusters on IBM i5/OS

10. Perform the following steps to access the Web Content Management content through an external Web server:
   A. Log on to the deployment manager administrative console.
   B. Select Environment > WebSphere Variables.
   C. From the Scope drop-down menu, select the Node=nodename, Server=servername option to narrow the scope of the listed variables, where Node=nodename is the node that contains the application server.
   D. Update the WCM_HOST variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.
   E. Update the WCM_PORT variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.
   F. Perform the following steps to synchronize the node with the deployment manager:
      1. Select System Administration > Nodes.
      2. Select the node that you want to synchronize from the list.
      3. Click Full Resynchronize.
   G. Log off of the deployment manager administrative console.

11. Run the following tasks to propagate your changes: **Note:** WebSphere_Portal_nodename is the name of the node’s WebSphere Portal server; but if you customized the server name, the default name changes. If you are uncertain of the server name, run the serverstatus -all task to get a list of the server names and their status.
   A. stopManager -username admin_userid -password admin_password, from the dmgr_profile_root/bin directory
   B. stopNode -username admin_userid -password admin_password from the wp_profile_root/bin directory
   C. stopServer WebSphere_Portal -username admin_userid -password admin_password
   D. startManager, from the dmgr_profile_root/bin directory
   E. startNode, from the wp_profile_root/bin directory
   F. startServer.sh WebSphere_Portal_nodename

12. Change the task list settings to point to the cluster instead of a single server if using Portal business process integration. See one of the following sections in the Information Center for information:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross cell</td>
<td>Adding a BPI-enabled portal server to a managed cell in a cross-cell setup</td>
</tr>
<tr>
<td>Single cell</td>
<td>Adding a BPI-enabled portal server to a managed cell in a single-cell setup</td>
</tr>
</tbody>
</table>

**Parent topic:** Setting up a cluster on IBM i5/OS

**Previous topic:** Preparing the primary node on IBM i5/OS

**Next topic:** Preparing the Web server when portal is installed on IBM i5/OS

**Related tasks**
- Adding a BPI-enabled portal server to a managed cell in a cross-cell setup
- Adding a BPI-enabled portal server to a managed cell in a single-cell setup
Preparing the Web server when portal is installed on IBM i5/OS

Install and configure the Web server plug-in, provided by IBM® WebSphere® Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

**Prerequisites**

- Preparing the WebSphere Application Server Deployment Manager on i5/OS
- Preparing prerequisite and corequisite software on IBM i5/OS
- Preparing your IBM i5/OS
- Preparing the primary node on IBM i5/OS

Perform the following steps to install and configure your Web server:

1. Install and configure the Web server; refer to the Web server documentation for information.
2. If using IBM Lotus® Domino®, edit the `NOTES.INI` file on the Web server. Set the `HTTPEnableConnectorHeaders` parameter to 1.
3. If using IBM HTTP Server or Apache Server, edit the `httpd.conf` file on the Web server. Set the `AllowEncodedSlashes` directive to `On`; the directive should be added at the root level as a global directive.
4. Stop the Web server.
5. Install and configure the Web server plug-in on the system where the Web server is located using the Plug-ins installation wizard provided with WebSphere Application Server. Refer to Selecting a Web server topology diagram and roadmap for information.

   **Important:** Depending on how you use the Web server, you may need to adjust the `ServerIOTimeout` value, which defines how long the plug-in should wait for a response from the application. The recommended minimum value is 60 but you may need to adjust this value higher if you are retrieving data from a database. To update this value, locate and open your `plugin-cfg.xml` file and set `AcceptAllContent` to `true`.

   **Important:** After successfully installing the Web server plug-in, locate and open your `plugin-cfg.xml` file and set `ServerIOTimeout` to a value that is appropriate for your business needs. For additional information, see Common questions about the Web server plug-in.

6. If you are using Sun Web Server Version 7 update 8, read Technote 1448262 and perform the steps to resolve the HTTP 408/409 error.
7. Start the Web server.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>See the appropriate HTTP Server documentation</td>
<td>IBM HTTP Server</td>
</tr>
<tr>
<td>See the appropriate Apache Server documentation</td>
<td><code>AllowEncodedSlashes</code> directives</td>
</tr>
</tbody>
</table>

**Parent topic:** Preparing prerequisite and corequisite software on IBM i5/OS

**Previous topic:** Preparing the WebSphere Application Server Deployment Manager on i5/OS

**Parent topic:** Setting up a cluster on IBM i5/OS

**Previous topic:** Creating the cluster on i5/OS

**Next topic:** Preparing user registries on IBM i5/OS
Preparing user registries on IBM i5/OS

Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on IBM i5/OS
  - Preparing your IBM i5/OS
  - Preparing the primary node on IBM i5/OS

Choose the appropriate LDAP server to install and setup:

- **Preparing a Tivoli Directory Server for IBM i5/OS**
  If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

Parent topic: Setting up a cluster on IBM i5/OS
Previous topic: Preparing the Web server when portal is installed on IBM i5/OS
Next topic: Configuring WebSphere Portal to use a user registry on IBM i5/OS in a clustered environment
Preparing a Tivoli Directory Server for IBM i5/OS

If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Tivoli Directory Server:

1. Customize the LDAP directory servers settings using the Directory Services Configuration Wizard. You must have *ALLOBJ and *IOSYSCFG special authority to use the wizard. Go to IBM System i and IBM i Information Center, select the appropriate Information Center version and navigate to e-business and Web serving > Security and IBM Tivoli Directory Server for i5/OS (LDAP) > IBM Tivoli Directory Server for i5/OS (LDAP) for information. **Note:** Due to a restriction in Tivoli Directory Server, users or groups must not contain a Turkish uppercase dotted I or lowercase dotted i in the DN as this will prevent correct retrieval of that user or group.

2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Go to IBM System i and IBM i Information Center, select the appropriate Information Center version and navigate to Networking > TCP/IP applications, protocols, and services > IBM Directory Server for iSeries (LDAP) > Administering Directory Server > General administration tasks > Adding and Removing Directory Server suffixes for information.
      2. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every dc=yourco,dc=com with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.
   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.

**Parent topic:** Preparing user registries on IBM i5/OS
Configuring WebSphere Portal to use a user registry on IBM i5/OS in a clustered environment

Configure user registry security on IBM® WebSphere® Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

Prior to configuring security, you should use the IBM WebSphere Application Server backupConfig task to create and store a backup of the IBM WebSphere Portal configuration; see backupConfig command for information.

- Prerequisites
  - Preparing prerequisite and corequisite software on IBM i5/OS
  - Preparing your IBM i5/OS
  - Preparing the primary node on IBM i5/OS

Perform the following tasks to configure WebSphere Portal to use a user registry:

1. Choosing your user registry model on IBM i5/OS in a clustered environment

Choose between securing IBM WebSphere Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

Parent topic: Setting up a cluster on IBM i5/OS
Previous topic: Preparing user registries on IBM i5/OS
Next topic: Preparing additional nodes on IBM i5/OS
Choosing your user registry model on IBM i5/OS in a clustered environment

Choose between securing IBM® WebSphere® Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

Choose one of the following user registry models:

- **Configuring a stand-alone LDAP user registry on IBM i5/OS in a clustered environment**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Configuring the default federated repository on IBM i5/OS in a clustered environment**
  Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

- **Adapting the attribute configuration**
  After installing IBM WebSphere Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

**Parent topic:** Configuring WebSphere Portal to use a user registry on IBM i5/OS in a clustered environment
Configuring a stand-alone LDAP user registry on IBM i5/OS in a clustered environment

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Configuring a stand-alone LDAP user registry on IBM i5/OS in a clustered environment**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry to store all user account information for authorization.

- **Configuring a stand-alone LDAP user registry over SSL on IBM i5/OS in a clustered environment**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

**Parent topic:** Choosing your user registry model on IBM i5/OS in a clustered environment
Configuring a stand-alone LDAP user registry on IBM i5/OS in a clustered environment

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry to store all user account information for authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps. If you need to rerun the `wp-modify-ldap-security` task to change the LDAP repositories or because the task failed, you must choose a new name for the realm using the `standalone.ldap.realm` parameter or you can set `ignoreDuplicateIDs=true` in the `wkplc.properties` file, before rerunning the task.

Perform the following steps to configure a standalone LDAP user registry:

1. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory. 
   - Required: Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Stand-alone LDAP configuration heading:
     - `standalone.ldap.id`
     - `standalone.ldap.host`
     - `standalone.ldap.port`
     - `standalone.ldap.bindDN`
     - `standalone.ldap.bindPassword`
     - `standalone.ldap.ldapServerType`
     - `standalone.ldap.userIdMap`
     - `standalone.ldap.groupIdMap`
     - `standalone.ldap.groupMemberIdMap`
     - `standalone.ldap.userFilter`
     - `standalone.ldap.groupFilter`
     - `standalone.ldap.serverId`
     - `standalone.ldap.serverPassword`
     - `standalone.ldap.realm`
     - `standalone.ldap.primaryAdminId`
     - `standalone.ldap.primaryAdminPassword`
     - `standalone.ldap.primaryPortalAdminId`
     - `standalone.ldap.primaryPortalAdminPassword`
     - `standalone.ldap.primaryPortalAdminGroup`
     - `standalone.ldap.baseDN`

   - `Note`: Use the `wp_security_xxx.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_security_xxx.properties` helper file.

2. Use the `wp_security_xxx.properties` helper file, located in the `wp_profile_root/ConfigEngine/properties` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_security_xxx.properties` helper file.
3. Required: Enter a value for the following required entity types parameters in the wkplc.properties file under the LDAP entity types heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.et.group.objectClasses
   - standalone.ldap.et.group.objectClassesForCreate
   - standalone.ldap.et.group.searchBases
   - standalone.ldap.et.personaccount.objectClasses
   - standalone.ldap.et.personaccount.objectClassesForCreate
   - standalone.ldap.et.personaccount.searchBases

4. Required: Enter a value for the following required group member parameters in the wkplc.properties file under the Group member attributes heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.gm.groupMemberName
   - standalone.ldap.gm.objectClass
   - standalone.ldap.gm.scope
   - standalone.ldap.gm dummyMember

5. Required: Enter a value for the following required relative distinguished name (RDN®) parameters in the wkplc.properties file under the Default parent, RDN attribute heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.ldap.personAccountParent
   - standalone.ldap.groupParent
   - standalone.ldap.personAccountRdnProperties
   - standalone.ldap.groupRdnProperties

6. Save your changes to the wkplc.properties file.

7. Optional: Enter the following Web content authors parameters in the wkplc_comp.properties file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. Note: See the wkplc_comp.properties file for specific information about the required parameters and for advanced parameters.
   - WcmContentAuthorsGroupId
   - WcmContentAuthorsGroupCN

8. Run the ConfigEngine.sh validate-standalone-ldap -DWasPassword= password task to validate your LDAP server settings. Attention: If you have not deleted the default file repository, WasPassword is the value entered during installation and not a value found in your LDAP user registry. Note: During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press y then Enter.

9. Run the ConfigEngine.sh wp-modify-ldap-security -DWasPassword= password task, from the wp_profile_root /ConfigEngine directory, to set the standalone LDAP user registry.

10. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

11. Run the ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword= password task, from the wp_profile_root /ConfigEngine directory, to check that all defined attributes are available in the configured LDAP user registry. Important: When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

12. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the
LDAP directory.

**Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the configure-express task.

A. Edit the `wp_profile_root`/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties file.

B. Add the following lines to the file:

```
uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN

cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
```

Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. *Value for realm_name when running the Member Fixer task to update the member names used by Web Content Management*

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone ldap realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

13. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring a stand-alone LDAP user registry on IBM i5/OS in a clustered environment

**Related tasks**

- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation
Configuring a stand-alone LDAP user registry over SSL on IBM i5/OS in a clustered environment

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to configure a standalone LDAP user registry over SSL: Note: Use the `wp_security_xxx.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_security_xxx.properties` helper file.

1. To specify the LDAP server's SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:
   - **Server trust store**
     A. Add the certificate to the trust store:

        1. Log in to the WebSphere Application Server Administrative Console.

        2. Navigate to Security > SSL certificate and key management > SSL configurations.

        3. Click the appropriate SSL configuration from the list. For example,

           - Stand-alone environments: `NodeDefaultSSLSettings`
           - Clustered environments: `CellDefaultSSLSettings`

        **Clustered environments:** Ensure the setting for SSL configuration for outbound connection matches your SSL settings.

        4. Click Key stores and certificates.

        5. Click the appropriate trust store from the list; for example, `NodeDefaultTrustStore`.

        6. Click Signer certificates, click Add, and then enter the following information:

           - Type the **Alias** the key store uses for the signer certificate.
           - Type the **File name** where the signer certificate is located.

        7. Click OK and then click Save to save the changes to the master configuration.

   B. Retrieve the certificate from the port:

       1. Log in to the WebSphere Application Server Administrative Console.

       2. Navigate to Security > SSL certificate and key management > SSL configurations.

       3. Click the appropriate SSL configuration from the list. For example,

           - Stand-alone environments: `NodeDefaultSSLSettings`
Clustered environments: **CellDefaultSSLSettings**

Clustered environments: Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.

4. Click **Key stores and certificates**.

5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.

6. Click **Signer certificates**, click **Retrieve from port**, and then enter the following information:
   - Type the **Host** name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the SSL **Port** used when attempting to retrieve the signer certificate.
   - Type the **Alias** the key store uses for the signer certificate.

7. Click **Retrieve signer information** to retrieve the certificate from the port.

8. Click **OK** and then click **Save** to save the changes to the master configuration.

- **Client trust storeNote**: This task might report an error, but it does successfully update the trust store. You can ignore the error message.

A. See **Secure installation for client signer retrieval**.

B. Run the **retrieveSigners** task from the `wp_profile_root\bin` directory; see **retrieveSigners command** for information. In a deployed environment, you will need to run the retrieveSigners task, for any federated node, against the Deployment Manager. **Note**: This task might report an error, but it does successfully update the trust store. You can ignore the error message.

C. Update the trust store properties file:

1. Use a text editor to open the `ssl.client.props` file, located in the `wp_profile_root\properties` directory.

2. Change the **com.ibm.ssl.trustStore** parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter

   ```
   com.ibm.ssl.trustStore=${CONFIG_ROOT}\cells\wpsbvt\nodes\wpsbvt\trust.p12
   ```

   to use the default trust store.

3. Save your changes.

2. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

3. **Required**: Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Standalone LDAP configuration heading: **Note**: See the properties file for specific information about the required parameters and for advanced parameters.

   - `standalone.idap.id`
   - `standalone.idap.host`
   - `standalone.idap.port`
   - `standalone.idap.blndDN`
   - `standalone.idap.bindPassword`
   - `standalone.idap.idapServerType`
   - `standalone.idap.userldMap`
   - `standalone.idap.groupIdMap`
   - `standalone.idap.groupMemberIdMap`
   - `standalone.idap.userFilter`
   - `standalone.idap.groupFilter`
   - `standalone.idap.serverId`
   - `standalone.idap.serverPassword`
   - `standalone.idap.realm`
   - `standalone.idap.primaryAdminId`
   - `standalone.idap.primaryAdminPassword`
   - `standalone.idap.primaryPortalAdminId`
4. Required: Enter a value for the following required entity types parameters in the \texttt{wkplc.properties} file under the LDAP entity types heading:\textbf{Note:} See the properties file for specific information about the required parameters and for advanced parameters.

- \texttt{standalone.ldap.et.group.objectClasses}
- \texttt{standalone.ldap.et.group.objectClassesForCreate}
- \texttt{standalone.ldap.et.group.searchBases}
- \texttt{standalone.ldap.et.personaccount.objectClasses}
- \texttt{standalone.ldap.et.personaccount.objectClassesForCreate}
- \texttt{standalone.ldap.et.personaccount.searchBases}

5. Required: Enter a value for the following required group member parameters in the \texttt{wkplc.properties} file under the Group member attributes heading:\textbf{Note:} See the properties file for specific information about the required parameters and for advanced parameters.

- \texttt{standalone.ldap.gm.groupMemberName}
- \texttt{standalone.ldap.gm.objectClass}
- \texttt{standalone.ldap.gm.scope}
- \texttt{standalone.ldap.gm.dummyMember}

6. Required: Enter a value for the following required relative distinguished name (RDN®) parameters in the \texttt{wkplc.properties} file under the Default parent, RDN attribute heading:\textbf{Note:} See the properties file for specific information about the required parameters and for advanced parameters.

- \texttt{standalone.ldap.personAccountParent}
- \texttt{standalone.ldap.groupParent}
- \texttt{standalone.ldap.personAccountRdnProperties}
- \texttt{standalone.ldap.groupRdnProperties}

7. Enter a value for the following parameters to enable Secure Socket Layers (SSL):\textbf{Note:} See the properties file for specific information about the required parameters and for advanced parameters.

Required parameters:

- \texttt{standalone.ldap.sslEnabled}
- \texttt{standalone.ldap.sslConfiguration}

Optional parameters:

- \texttt{standalone.ldap.certificateMapMode}
- \texttt{standalone.ldap.certificateFilter}

8. Save your changes to the \texttt{wkplc.properties} file.

9. Run the \texttt{ConfigEngine.sh \ validate-standalone-ldap -DWasPassword=\textit{password}} task to validate your LDAP server settings.\textbf{Attention:} If you have not deleted the default file repository, \textit{WasPassword} is the value entered during installation and not a value found in your LDAP user registry. 

\textbf{Note:} During the validation task, you may receive the following prompt: Add signer to the trust store now? Press \texttt{y} then \texttt{Enter}.

10. Run the \texttt{ConfigEngine.sh \ wp-modify-ldap-security -DWasPassword=\textit{password}} task, from the \texttt{wp_profile_root/ConfigEngine} directory, to set the standalone LDAP user registry.

11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: \textit{Starting and stopping servers, deployment managers, and node agents}.

12. Run the \texttt{ConfigEngine.sh \ wp-validate-standalone-ldap-attribute-config -DWasPassword=\textit{password}} task, from the \texttt{wp_profile_root/ConfigEngine} directory, to check that all defined attributes are available in the configured
LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring a stand-alone LDAP user registry on IBM i5/OS in a clustered environment

**Related tasks**
- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation
Configuring the default federated repository on IBM i5/OS in a clustered environment

Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

Perform the following tasks to add user registries to the default federated repository:

- **Configuring a federated LDAP user registry on IBM i5/OS in a clustered environment**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Adding a database user registry on IBM i5/OS in a clustered environment**
  Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

- **Adding realm support on IBM i5/OS in a clustered environment**
  A realm is a group of users from one or more user registries that form a coherent group within IBM WebSphere Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

**Parent topic:** Choosing your user registry model on IBM i5/OS in a clustered environment
Configuring a federated LDAP user registry on IBM i5/OS in a clustered environment

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- Adding an LDAP user registry on IBM i5/OS in a clustered environment
  Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time. If IBM Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=groupName` and the hierarchical format is `cn=groupName,o=root`. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as `wpsadmin`, this ID cannot exist in the LDAP you are adding.

- Adding an LDAP user registry over SSL on IBM i5/OS in a clustered environment
  Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

**Parent topic:** Configuring the default federated repository on IBM i5/OS in a clustered environment
Adding an LDAP user registry on IBM i5/OS in a clustered environment

Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time. If IBM® Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is \texttt{cn=groupName} and the hierarchical format is \texttt{cn=groupName,o=\textit{root}}. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as \texttt{wpsadmin}, this ID cannot exist in the LDAP you are adding.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

\textbf{Tip:} Perform these steps on the primary node only.

\textbf{Note:} Use the \texttt{wp_add_federated\_xxx.properties} helper file, located in the \texttt{wp\_profile\_root/ConfigEngine/config/helpers} directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the \texttt{wkplc.properties} file, you will use your \texttt{wp\_add\_federated\_xxx.properties} helper file.

1. Use a text editor to open the \texttt{wkplc.properties} file, located in the \texttt{wp\_profile\_root/ConfigEngine/properties} directory.

2. Required: Enter a value for the following required parameters in the \texttt{wkplc.properties} file under the VMM Federated LDAP Properties heading: \textbf{Note:} See the properties file for specific information about the required parameters and for advanced parameters.

   - \texttt{federated.ldap.id}
   - \texttt{federated.ldap.host}
   - \texttt{federated.ldap.port}
   - \texttt{federated.ldap.bindDN}
   - \texttt{federated.ldap.bindPassword}
   - \texttt{federated.ldap.ldapServerType}
   - \texttt{federated.ldap.baseDN}

3. Required: Enter a value for the following required entity types parameters in the \texttt{wkplc.properties} file under the LDAP entity types heading: \textbf{Note:} See the properties file for specific information about the required parameters and for advanced parameters.

   - \texttt{federated.ldap.et.group.objectClasses}
   - \texttt{federated.ldap.et.group.objectClassesForCreate}
   - \texttt{federated.ldap.et.group.searchBases}
   - \texttt{federated.ldap.et.personaccount.objectClasses}
4. Required: Enter a value for the following required group member parameters in the \texttt{wkplc.properties} file under the Group member attribute heading:\textbf{Note:} See the properties file for specific information about the required parameters and for advanced parameters.
- \texttt{federated.ldap.et.personaccount.objectClassesForCreate}
- \texttt{federated.ldap.et.personaccount.searchBases}
- \texttt{federated.ldap.et.personaccount.objectClassesForCreate}
- \texttt{federated.ldap.et.personaccount.objectClassesForCreate}
- \texttt{federated.ldap.et.personaccount.searchBases}

5. Save your changes to the \texttt{wkplc.properties} file.

6. Optional: Enter the following Web content authors parameters in the \texttt{wkplc_comp.properties} file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product.\textbf{Note:} See the \texttt{wkplc_comp.properties} file for specific information about the required parameters and for advanced parameters.
- \texttt{WcmContentAuthorsGroupId}
- \texttt{WcmContentAuthorsGroupCN}

7. Run the \texttt{ConfigEngine.sh validate-federated-ldap -DWasPassword=password} task to validate your LDAP server settings.\textbf{Attention:} If you have not deleted the default file repository, \texttt{WasPassword} is the value entered during installation and not a value found in your LDAP user registry.

\textbf{Note:} During the validation task, you may receive the following prompt: \texttt{Add signer to the trust store now?}. Press \texttt{y} then \texttt{Enter}.

8. Run the \texttt{ConfigEngine.sh wp-create-ldap -DWasPassword=password} task, from the \texttt{wp_profile_root/ConfigEngine} directory, to add an LDAP user registry to the default federated repository.\textbf{Note:} Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: \textit{Starting and stopping servers, deployment managers, and node agents}.

10. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:

A. Use a text editor to open the \texttt{wkplc.properties} file, located in the \texttt{wp_profile_root/ConfigEngine/properties} directory.

B. Enter a value for the following required parameters in the \texttt{wkplc.properties} file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms:

\textbf{Note:} See the properties file for specific information about the required parameters and for advanced parameters.
- \texttt{id}
- \texttt{baseDN}
- \texttt{nameInRepository}

C. Save your changes to the \texttt{wkplc.properties} file.

D. Run the \texttt{ConfigEngine.sh wp-create-base-entry -DWasPassword=password} task, from the \texttt{wp_profile_root/ConfigEngine} directory, to create a base entry in a repository.

E. Stop and restart all necessary servers to propagate your changes.

11. Optional: Run the \texttt{ConfigEngine.sh wp-query-repository -DWasPassword=password} task, from the \texttt{wp_profile_root/ConfigEngine} directory, to list the names and types of configured repositories.

12. Run the \texttt{ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password} task, from the \texttt{wp_profile_root/ConfigEngine} directory, to check that all defined attributes are available in the configured LDAP.
user registry. **Important:** When you finish configuring your LDAP user registry, see “Adapting the attribute configuration” for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

13. Perform the following steps to update the user registry where new users and groups are stored:

   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - `personAccountParent`
      - `groupParent`
      - `personAccountRdnProperties`
      - `groupRdnProperties`

      The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:

      - `personAccountParent=dc=yourco,dc=com`
      - `groupParent=dc=yourco,dc=com`

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to delete the old attributes before adding the new attributes.

   E. Stop and restart all necessary servers to propagate your changes.

14. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for `realmName` or leave blank to update the default realm.

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

   E. Stop and restart all necessary servers to propagate your changes.

15. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

   **Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

   A. Edit the `wp_profile_root/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties` file.

   B. Add the following lines to the file:

   ```
   uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
   cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
   ```
Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management.

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

16. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

17. Optional: **This step is required in a production environment.** Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

**Important:**
- Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
- Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

**Note:** If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

A. Run the following task from the `wp_profile_root/ConfigEngine` directory: `ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword`. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` parameter.

B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

C. Run the following task from the `wp_profile_root/ConfigEngine` directory: `ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid`. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `Dskip.ldap.validation=true` parameter to the task to skip the validation.

D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.
18. Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it.

   The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: *Deleting the repository.*

   If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring a federated LDAP user registry on IBM i5/OS in a clustered environment

**Related tasks**

Starting and stopping servers, deployment managers, and node agents

Enabling LDAP security after cluster creation

**Related information**

User IDs and passwords
Adding an LDAP user registry over SSL on IBM i5/OS in a clustered environment

Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry over SSL to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

**Tip:** Perform these steps on the primary node only.

**Note:** Use the wp_add_federated_xxx.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_add_federated_xxx.properties helper file.

To specify the LDAP server’s SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:

- **Server trust store**
  A. Add the certificate to the trust store:
    1. Log in to the WebSphere Application Server Administrative Console.
    2. Navigate to Security > SSL certificate and key management > SSL configurations.
    3. Click the appropriate SSL configuration from the list. For example,
      - **Stand-alone environments:** NodeDefaultSSLSettings
      - **Clustered environments:** CellDefaultSSLSettings
    4. **Clustered environments:** Ensure the setting for SSL configuration for outbound connection matches your SSL settings.
    5. Click Key stores and certificates.
    6. Click Signer certificates, click Add, and then enter the following information:
      - Type the Alias the key store uses for the signer certificate.
      - Type the File name where the signer certificate is located.
    7. Click OK and then click Save to save the changes to the master configuration.

- **B. Retrieve the certificate from the port:**
  1. Log in to the WebSphere Application Server Administrative Console.
2. Navigate to **Security > SSL certificate and key management > SSL configurations**.

3. Click the appropriate SSL configuration from the list. For example,
   - Stand-alone environments: **NodeDefaultSSLSettings**
   - Clustered environments: **CellDefaultSSLSettings**

   **Clustered environments:** Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.

4. Click **Key stores and certificates**.

5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.

6. Click **Signer certificates**, click **Retrieve from port**, and then enter the following information:
   - Type the **Host** name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the SSL **Port** used when attempting to retrieve the signer certificate.
   - Type the **Alias** the key store uses for the signer certificate.

7. Click **Retrieve signer information** to retrieve the certificate from the port.

8. Click **OK** and then click **Save** to save the changes to the master configuration.

   **Client trust store**
   
   **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

   A. See Secure installation for client signer retrieval.

   B. Run the `retrieveSigners` task from the `wp_profile_root\bin` directory; see `retrieveSigners command` for information. In a deployed environment, you will need to run the `retrieveSigners` task, for any federated node, against the Deployment Manager.

   **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.

C. Update the trust store properties file:

   1. Use a text editor to open the `ssl.client.props` file, located in the `wp_profile_root\properties` directory.

   2. Change the `com.ibm.ssl.trustStore` parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter
      
      ```
      com.ibm.ssl.trustStore=${CONFIG_ROOT}\cells\wpsbvt\nodes\wpsbvt\trust.p12
      ```
      
      to use the default trust store.

   3. Save your changes.

2. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

3. Required: Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated LDAP Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - `federated.idap.id`
   - `federated.idap.host`
   - `federated.idap.port`
   - `federated.idap.blindDN`
   - `federated.idap.blindPassword`
   - `federated.idap.idapServerType`
   - `federated.idap.baseDN`

4. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - `federated.idap.et.group.objectClasses`
   - `federated.idap.et.group.objectClassesForCreate`
   - `federated.idap.et.group.searchBases`
- `federated.ldap.et.personaccount.objectClasses`
- `federated.ldap.et.personaccount.objectClassesForCreate`
- `federated.ldap.et.personaccount.searchBases`

5. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attribute heading:
   - `federated.ldap.gm.groupMemberName`
   - `federated.ldap.gm.objectClass`
   - `federated.ldap.gm.scope`
   - `federated.ldap.gm.dummyMember`

   Note: See the properties file for specific information about the required parameters and for advanced parameters.

6. Enter a value for the following parameters to enable Secure Socket Layers (SSL):
   - `federated.ldap.sslEnabled`
   - `federated.ldap.sslConfiguration`

   Required parameters:
   - `federated.ldap.certificateMapMode`
   - `federated.ldap.certificateFilter`

   Optional parameters:
   - `federated.ldap.certificateMapMode`
   - `federated.ldap.certificateFilter`

7. Save your changes to the `wkplc.properties` file.

8. Optional: Enter the following Web content authors parameters in the `wkplc_comp.properties` file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product:
   - `WcmContentAuthorsGroupId`
   - `WcmContentAuthorsGroupCN`

   Note: See the `wkplc_comp.properties` file for specific information about the required parameters and for advanced parameters.

9. Run the `ConfigEngine.sh validate-federated-ldap -DWasPassword=passw...ted repository. Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks:
    - Starting and stopping servers, deployment managers, and node agents.

12. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:
   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms:
      - `Note: See the properties file for specific information about the required parameters and for advanced parameters.`
C. Save your changes to the wkplc.properties file.

D. Run the `ConfigEngine.sh wp-create-base-entry -DWasPassword=password` task, from the `wp_profile_root` /ConfigEngine directory, to create a base entry in a repository.

E. Stop and restart all necessary servers to propagate your changes.

13. Optional: Run the `ConfigEngine.sh wp-query-repository -DWasPassword=password` task, from the `wp_profile_root` /ConfigEngine directory, to list the names and types of configured repositories.

14. Run the `ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root`/ConfigEngine directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see “Adapting the attribute configuration” for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

15. Perform the following steps to update the user registry where new users and groups are stored:

   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root`/ConfigEngine/properties directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

   **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - `personAccountParent`
      - `groupParent`
      - `personAccountRdnProperties`
      - `groupRdnProperties`

   The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:

      - `personAccountParent=dc=yourco,dc=com`
      - `groupParent=dc=yourco,dc=com`

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root` /ConfigEngine directory, to delete the old attributes before adding the new attributes.

   E. Stop and restart all necessary servers to propagate your changes.

16. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root`/ConfigEngine/properties directory.

   B. Enter a value for `realmName` or leave blank to update the default realm.

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root`/ConfigEngine directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.
E. Stop and restart all necessary servers to propagate your changes.

17. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

**Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

A. Edit the `/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties` file.

B. Add the following lines to the file:

```
uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
```

Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `ConfigEngine.sh action-express-memberfixer` task, located in the `/PortalServer/wcm/shared/app/config/wcmservices/` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the wkplc.properties file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the wkplc.properties file. If the value for <code>federated.realm</code> is empty, use defaultWIMFileBasedRealm as the default value.</td>
</tr>
</tbody>
</table>

18. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

19. Optional: **This step is required in a production environment.** Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

**Important:**
- Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
- Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

**Note:** If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

A. Run the following task from the `/PortalServer/wcm/shared/app/config/wcmservices/` directory: `ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword` **Important:** You must provide the full distinguished name (DN) for the `newAdminId` parameter.
B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

C. Run the following task from the \( wp\text{-profile}\text{root}/\text{ConfigEngine} \) directory: \texttt{ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid}. \textbf{Important:} You must provide the full distinguished name (DN) for the \texttt{newAdminId} and \texttt{newAdminGroupId} parameters.

\textbf{Additional parameter for stopped servers:} This task verifies the user against a running server instance. If the server is stopped, add the \texttt{-Dskip.ldap.validation=true} parameter to the task to skip the validation.

D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

20. Optional: \textbf{This step is required in a production environment.} Remove the file system repository if you do not use it. The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: \textit{Deleting the repository}.

If you created your clustered environment then performed the steps in this task, you must now run the \texttt{enable-jcr-security} task on the secondary node. See \textit{Enabling LDAP security after cluster creation} for instructions.

\textbf{Parent topic:} Configuring a federated LDAP user registry on IBM i5/OS in a clustered environment

\textbf{Related tasks}
Starting and stopping servers, deployment managers, and node agents
Enabling LDAP security after cluster creation

\textbf{Related Information}
User IDs and passwords
Adding a database user registry on IBM i5/OS in a clustered environment

Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps. If you have WebSphere Application Server Version 7.0.x installed, you must install APAR PM23090 and APAR PM24181 for WebSphere Portal prior to running this task. Clusters with WebSphere Application Server Version 6.1.x do not require these APARs.

Perform the following steps to add a database user registry to the default federated repository; you must repeat these steps for each additional database user registry that you plan to add:

**Tip:** Perform these steps on the primary node only.

**Note:** Use the `wp_add_DB.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_add_DB.properties` helper file.

1. Perform the following steps to create the DB2 for i5/OS database:

**Instructions for setting up databases:** Refer to the appropriate documentation for the type of database you want to set up.

**Consulting your database administrator:** The task of setting up a new database is typically performed by a database administrator. However, the following steps are provided for your reference in the event you are creating a stand-alone database for testing or demonstration purposes. Consult your database administrator before proceeding with the following steps if you plan to create a database for a production environment.

   A. Login to a remote i5/OS session.
   B. Enter the `strsql` command to start the interactive sql session.
   C. Enter the `create schema database_name` command, where `database_name` is the name you want to use for the database.

2. Perform the following steps to define the `DbDriver` and `DbLibrary` parameter values:

   A. Navigate to the following directory: `wp_profile_root/ConfigEngine/properties` directory.
   B. Locate and open `wkplc_dbtype.properties` with any text editor.
   C. Enter a value for the following parameters under the appropriate database type properties heading:
      - `db_type.DbDriver`
      - `db_type.DbLibrary`
   D. Save your changes.

3. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

4. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated Database Properties heading:

   **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
5. Change the value for the `com.ibm.SOAP.requestTimeout` parameter to 1000.
   A. Navigate to the following directory: `wp_profile_root/properties`
   B. Locate and open `soap.client.props` with any text editor.
   C. Locate the `com.ibm.SOAP.requestTimeout` parameter and change the value to 1000.
   D. Save and close `soap.client.props`.

6. Perform the following steps in a clustered environment:
   A. Run the `ConfigEngine.sh wp-prep-vmm-db-secured-environment -DWasPassword=password -` 
      `DDbDomain=federated.db -Db_type.DmgrDbLibrary=local path of the database jars on the Deployment` 
      `Manager -DDmgrNodeName=dmgr_node_name` task from the `wp_profile_root/ConfigEngine` directory to create the local 
      Deployment Manager WebSphere variable used to access the database jars. **Note:** The `db_type` in `db_type` 
      `.DmgrDbLibrary` should be set to the type of database you are using, for example `db2_iseseries`. The `local path of` 
      `the database jars on the Deployment Manager` should be one of the following options:

      `DB2 for i5/OS Type 2 driver:/QIBM/ProdData/Java400/ext/db2_classes.jar`
      `DB2 for i5/OS Type 4 driver:/QIBM/ProdData/HTTP/Public/jt400/lib/jt400.jar`

   B. Run the following task. Include each node name as a comma separated list in the command:
      **Running the task:** You do not have to run this task more than once. You can run this task from any node in the
      cluster.

      1. Set the property value for `federated.db.DbType` if using a database user registry and set the property value for
         `la.DbType` if using a property extension database in the wkplc.properties file.
      2. Run the `ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=password -` 
         `DDbDomain=federated.db -DVmmNodeName=node_name -Db_type.NodeDbLibrary=local full path of the` 
         `database jars` task from the `wp_profile_root/ConfigEngine` directory on each node to create the variable used to
         access the VMM database jars. **Note:** `VmmNodeName` is a list of one or more WebSphere Portal nodes names in
         the cell which share the same database driver paths. The `db_type` in `db_type.NodeDbLibrary` should be set to
         the type of database you are using, for example `db2`.

         `DB2 for i5/OS Type 2 driver:/QIBM/ProdData/Java400/ext/db2_classes.jar`
         `DB2 for i5/OS Type 4 driver:/QIBM/ProdData/HTTP/Public/jt400/lib/jt400.jar`

   C. Stop and restart all necessary servers to propagate your changes.

   7. Run the `ConfigEngine.sh wp-create-db -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` 
      directory, to add a database user registry to the default federated repository. **Note:** Users who are not in an LDAP do 
      not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and 
      then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who 
      sign up using the Self Care portlet do not have awareness.
8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

9. Perform the following steps to update the user registry where new users and groups are stored:

   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - `personAccountParent`
      - `groupParent`
      - `personAccountRdnProperties`
      - `groupRdnProperties`

      The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:

      - `personAccountParent=dc=yourco,dc=com`
      - `groupParent=dc=yourco,dc=com`

      C. Save your changes to the `wkplc.properties` file.

      D. Run the `ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to delete the old attributes before adding the new attributes.

      E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Run the `ConfigEngine.sh wp-query-repository -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to list the names and types of configured repositories.

    If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See Enabling LDAP security after cluster creation for instructions.

**Parent topic:** Configuring the default federated repository on IBM i5/OS in a clustered environment

**Related tasks**

Starting and stopping servers, deployment managers, and node agents

Enabling LDAP security after cluster creation
Adding realm support on IBM i5/OS in a clustered environment

A realm is a group of users from one or more user registries that form a coherent group within IBM® WebSphere® Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

Before configuring realm support, you must add all LDAP user registries and/or database user registries, that you will use to create a single realm or multiple realms, to the federated repository. If you are going to create multiple realms, you must create all required base entries within your LDAP user registries and/or database user registries. All base entry names must be unique within the federated repository. In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add realm support to your user registry model:

1. **Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.**
   - **Required:** Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading:
     - realmName
     - securityUse
     - delimiter
     - addBaseEntry
   - **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

2. **Save your changes to the wkplc.properties file.**

3. **Run the ConfigEngine.sh wp-create-realm -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add a new realm to the Virtual Member Manager configuration.**
   - **Important:** To create multiple realms, ensure that your federated repository contains the required unique base entries. Stop and restart the appropriate servers for your installation environment, and then update the wkplc.properties file with the base entry information and rerun the wp-create-realm task. Repeat these steps until all realms are created.

4. **Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.**

5. **Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading and then save your changes.**
   - **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
     - realmName
     - realm.personAccountParent
     - realm.groupParent
     - realm.orgContainerParent

6. **Run the ConfigEngine.sh wp-modify-realm-defaultparents -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to update the default parents per entity type and realm.**
   - **Important:** Stop and restart the appropriate servers for your installation environment before rerunning this task for any additional entity types.
and realms.

8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

9. Optional: Perform the following steps to add additional base entries to the realm configuration; for example, if you had two additional base entries (base entry 1 and base entry 2) to add to the realm you just created, you would update the wkplc.properties file with the information from base entry 1 and then run this task. Then you would update the properties file with the information for base entry 2 and then run this task:
   A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
   B. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading:
      - realmName
      - addBaseEntry
   C. Save your changes to the wkplc.properties file.
   D. Run the ConfigEngine.sh wp-add-realm-baseentry -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add additional LDAP base entries to the realm configuration.
   E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID; this step is required if you change the default realm:
    A. Create a new user in the Manage Users and Groups portlet to replace the current WebSphere Application Server administrative user.
    B. Create a new user in the Manage Users and Groups portlet to replace the current WebSphere Portal administrative user.
    C. Create a new group in the Manage Users and Groups portlet to replace the current group.
    D. Run the ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid task, from the wp_profile_root/ConfigEngine directory, to replace the old WebSphere Application Server administrative user ID and group ID with the new user and group. Important: You must provide the full distinguished name (DN) for the newAdminId and newAdminGroupId parameters.

    Additional parameter for stopped servers: This task verifies the user against a running server instance. If the server is stopped, add the -Dskip.ldap.validation=true parameter to the task to skip the validation.
    E. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.
    F. Run the ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid task to replace the old WebSphere Portal administrative user ID and group ID with the new user and group. Important: You must provide the full distinguished name (DN) for the newAdminId and newAdminGroupId parameters.

    Additional parameter for stopped servers: This task verifies the user against a running server instance. If the server is stopped, add the -Dskip.ldap.validation=true parameter to the task to skip the validation.
    G. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

11. Optional: Perform the following steps to set the realm you created as the default realm: Remember: Only users defined in base entries that exist in the default realm are able to log into WebSphere Portal. If you find that a user cannot log in to WebSphere Portal, check to see if the base entry that contains the user exists in the default realm. You can run the wp-query-realm-baseentry task to see what base entries are part of the default realm. If the default realm is missing
the base entry, run the `wp-add-realm-baseentry` task to add the base entry to the default realm.
A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
B. For `defaultRealmName`, type the `realmName` property value you want to use as the default realm.
C. Save your changes to the `wkplc.properties` file.
D. Run the `ConfigEngine.sh wp-default-realm -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to set this realm as the default realm.
E. Stop and restart all necessary servers to propagate your changes.

12. Optional: Perform the following steps to query a realm for a list of its base entries:
A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
B. For `realmName`, type the name of the realm you want to query.
C. Save your changes to the `wkplc.properties` file.
D. Run the `ConfigEngine.sh wp-query-realm-baseentry -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to list the base entries for a specific realm.

13. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:
A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
B. Enter a value for `realmName` or leave blank to update the default realm.
C. Save your changes to the `wkplc.properties` file.
D. Run the `ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.
E. Stop and restart all necessary servers to propagate your changes.

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring the default federated repository on IBM i5/OS in a clustered environment

**Related tasks**

Starting and stopping servers, deployment managers, and node agents

Enabling LDAP security after cluster creation
Adapting the attribute configuration

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

After installation, IBM WebSphere Portal has a predefined set of attributes for users and groups. Your LDAP server may have a different set of predefined user and group attributes. To ensure proper communication between WebSphere Portal and your LDAP server, you can configure additional attributes and flag existing attributes as required or unsupported on a per repository basis or for all configure repositories.

LDAP servers can only handle attributes that are explicitly defined in their schema. The LDAP server’s schema is different from the WebSphere Portal schema but the two schemas should match for proper communication between WebSphere Portal and the LDAP server. The task to add the LDAP user registry does some basic attribute configurations depending on the type of LDAP server that you choose. You may, however, still need to adapt the WebSphere Portal configuration to match the LDAP schema; for example, if an attribute is defined in WebSphere Portal but not in the LDAP server, you will need to perform one of the following tasks to resolve this mismatch:

- Flag the attribute as unsupported for the LDAP server
- Introduce an attribute mapping that maps the WebSphere Portal attribute to an attribute defined in the LDAP schema

Perform the following tasks to adapt the attribute configuration to match the configured LDAP server(s) and your business needs:

1. **Querying the defined attributes on IBM i5/OS**
   After installing IBM WebSphere Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

2. **Adding attributes on IBM i5/OS in a clustered environment**
   The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM WebSphere Portal and your user registry.

3. **Mapping attributes on IBM i5/OS in a clustered environment**
   After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

4. **Removing attributes**
   Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

Parent topic: Choosing your user registry model on IBM i5/OS in a clustered environment

Related tasks
- Adding an LDAP user registry on IBM i5/OS
- Adding an LDAP user registry over SSL on IBM i5/OS
- Configuring a stand-alone LDAP user registry on IBM i5/OS
Configuring a stand-alone LDAP user registry over SSL on IBM i5/OS
Querying the defined attributes on IBM i5/OS

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

**Warning:** If you are using a database user registry or a property extension database, copy the database drivers to the `AppServer_root/lib` directory before executing this script.

Run the `ConfigEngine.sh wp-query-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, any time during the configuration process or at runtime to query an overview of the currently defined attributes. This task creates the `availableAttributes.html` report, located in the `wp_profile_root/ConfigEngine/log` directory. The report contains one table that lists the available attributes for Users (**PersonAccount**) and one table that lists the available attributes for Groups. For each configured repository there is a column that indicates if the attribute is flagged as unsupported or if the attribute is mapped to a different LDAP attribute. **Note:** This task does not validate the existence of attributes in the LDAP schema.

**Parent topic:** Adapting the attribute configuration

**Next topic:** Adding attributes on IBM i5/OS in a clustered environment
**Adding attributes on IBM i5/OS in a clustered environment**

The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM® WebSphere® Portal and your user registry.

Perform the following steps to add new attributes to your user registry:

1. Install the required Enterprise Archive (.ear) file on WebSphere Application Server. Select the appropriate task to install the .ear file:

   **Table 1. Steps for installing the .ear file by environment**

<table>
<thead>
<tr>
<th>Environment</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone environment</td>
<td>Perform the following steps to install the .ear file in a stand-alone environment: Open a command prompt. Navigate to the wp_profile_root/ConfigEngine directory. Run the ConfigEngine.sh wp-la-install-ear -DWasPassword=password task.</td>
</tr>
<tr>
<td>Clustered environment</td>
<td>Perform the following steps to install the .ear file in a clustered environment: Open a command prompt. Navigate to the wp_profile_root/ConfigEngine directory on the primary node. Run the ConfigEngine.sh wp-la-install-ear -DWasPassword=dmgr_password -DServerName=dmgr -DNodeName=node_name task. Where node_name is the name of the node where the deployment manager resides; you can find the node_name value in the WebSphere Application Server Administrative Console under <strong>System administrator &gt; Deployment Manager &gt; Runtime tab &gt; General Properties &gt; Node Name</strong>.</td>
</tr>
</tbody>
</table>

2. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

3. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

4. Enter a value for the following required parameters in the wkplc.properties file under the VMM Property Extension Properties heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.

   - la.providerURL
   - la.propertyName
   - la.entityTypes
   - la.dataType
   - la.multiValued

5. Save your changes to the wkplc.properties file.
6. Run the `ConfigEngine.sh wp-add-property -DWasPassword=password` task to add the attribute to the user registry.

**Note**: This task performs an EJB call to WebSphere Application Server, which must authenticate against WebSphere Application Server. Depending on the configuration in the `sas.client.props` file, you may receive a popup window or a command line prompt asking for user identity and password. Enter the WebSphere Application Server user ID and password.

**Remember**: If you have multiple properties to add, repeat all steps, except for the `wp-la-install-ear` task, until all new attributes are added.

7. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents*.

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Adapting the attribute configuration

**Previous topic:** Querying the defined attributes on IBM i5/OS

**Next topic:** Mapping attributes on IBM i5/OS in a clustered environment

**Related tasks**

Starting and stopping servers, deployment managers, and node agents

Enabling LDAP security after cluster creation
Mapping attributes on IBM i5/OS in a clustered environment

After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

Perform the following steps to map attributes between WebSphere Portal and your LDAP server; if you have multiple LDAP servers, you will need to perform these steps for each LDAP server:

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
2. Enter a value for one of the following sets of parameters in the wkplc.properties file to identify your LDAP server:
   - **Note:** Make sure you use the same values you used to configure your LDAP server.
   
   **Table 1. Identifying your LDAP server in the wkplc.properties file.**

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>The following parameters are found under the LDAP attribute configuration heading:<strong>Note:</strong> See the properties file for specific information about the required parameters and for advanced parameters. standalone.ldap.id standalone.ldap.host standalone.ldap.port standalone.ldap.sslEnabled standalone.ldap.bindDN standalone.ldap.bindPassword standalone.ldap.baseDN</td>
</tr>
<tr>
<td>Federated</td>
<td>The following parameters are found under the VMM Federated repository properties heading:<strong>Note:</strong> See the properties file for specific information about the required parameters and for advanced parameters. federated.ldap.id federated.ldap.host federated.ldap.port federated.ldap.sslEnabled federated.ldap.bindDN federated.ldap.bindPassword federated.ldap.baseDN</td>
</tr>
</tbody>
</table>

3. Run one of the following tasks to check that all defined attributes are available in the configured LDAP user registry:
   - **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   
   **Table 2. Task to check that all defined attributes are available in the configured LDAP user registry.**

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory.</td>
</tr>
<tr>
<td>Federated</td>
<td>ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory.</td>
</tr>
</tbody>
</table>

4. Open the ConfigTrace.log file, located in the wp_profile_root/ConfigEngine/log directory, to review the following output for the **PersonAccount** and **Group** entity type:
- The following attributes are defined in WebSphere Portal but not in the LDAP server
  - This list contains all attributes that are defined in WebSphere Portal but not available in the LDAP. Flag attributes that you do not plan to use in WebSphere Portal as unsupported. Map the attributes that you plan to use to the attributes that exist in the LDAP; you must also map the `uid`, `cn`, `firstName`, `sn`, `preferredLanguage`, and `ibm-primaryEmail` attributes if they are contained in the list.

- The following attributes are flagged as required in the LDAP server but not in WebSphere Portal
  - This list contains all attributes that are defined as "MUST" in the LDAP server but not as required in WebSphere Portal. You should flag these attributes as required within WebSphere Portal; see the step below about flagging an attribute as either unsupported or required.

- The following attributes have a different type in WebSphere Portal and in the LDAP server
  - This list contains all attributes that WebSphere Portal might ignore because the data type within WebSphere Portal and within the LDAP server do not match.

5. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

6. Enter a value for one of the following sets of parameters in the `wkplc.properties` file to correct any issues found in the config trace file: Table 3. Parameters to define in the wkplc.properties file to correct any issues found in the config trace file.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Parameters</th>
</tr>
</thead>
</table>
| Stand-alone     | The following parameters are found under the LDAP attribute configuration heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.  
|                 | `standalone.ldap.idstandalone.ldap.attributes.nonSupportedstandalone.ldap.attributes.mapping.ldapName=mail, title`  
|                 | `standalone.ldap.attributes.mapping.portalName=ibm-primaryEmail, ibm-jobTitle`  
|                 | `standalone.ldap.attributes.mapping.entityTypes=PersonAccount, Group` |
7. Save your changes to the wkplc.properties file.

8. Run one of the following tasks to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between WebSphere Portal and the LDAP user registry:

   Table 4. Task to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between Portal and the LDAP user registry.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>ConfigEngine.sh wp-update-standalone-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory</td>
</tr>
<tr>
<td>Federated</td>
<td>ConfigEngine.sh wp-update-federated-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory</td>
</tr>
</tbody>
</table>

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

10. Optional: Perform the following steps to flag an attribute as either unsupported or required for the entire WebSphere Portal environment instead of just for the specified LDAP:

    A. Enter a value for the following required parameters in the wkplc.properties file: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

       - `user.attributes.required`
       - `user.attributes.nonsupported`

    B. Save your changes to the wkplc.properties file.

    C. Run the `ConfigEngine.sh wp-update-attribute-config` -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory.

    D. Stop and restart all necessary servers to propagate your changes.

If you created your clustered environment then performed the steps in this task, you must now run the enable-jcr-security task on the secondary node. See Enabling LDAP security after cluster creation for instructions.
Removing attributes

Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

Perform the following steps to remove an attribute from your database: **Important:** Do not remove attributes that have already been populated with user values because this can cause database inconsistencies.

**Cluster note:** In a clustered environment, perform the following steps on the deployment manager and then resynch the nodes.

1. Perform the following steps to remove an attribute stored in a property extension database; this is an attribute that was added using the `wp-add-la-property` task.
   A. Open the tool you use to edit your database.
   B. Verify that your attribute name is available in the LAPROP table.
   C. Delete the required attributes from the LAPROP table.
   D. Open the `wimxmlextenstaion.xml` file, located in the `wp_profile_root/config/cells/cellname/wim/model` directory.
   E. Locate and delete the `propertySchema` definition for the attributes that you deleted from the LAPROP table; for example:

   ```xml
   <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String" multiValued="true" propertyName="attribute_name">
     <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
   </wim:propertySchema>
   ```
   F. Save your changes to the `wimxmlextenstaion.xml` file.
   G. Open the `wimconfig.xml` file, located in the `wp_profile_root/config/cells/cellname/wim/config` directory.
   H. Locate and delete the `propertiesNotSupported` definitions for the attributes that you deleted from the LAPROP table; for example:

   ```xml
   <config:propertiesNotSupported name="attribute_name"/>
   ```
   I. Save your changes to the `wimconfig.xml` file.
   J. Stop and restart the server1 and WebSphere_Portal servers from the `wp_profile_root/bin` directory.

2. Perform the following steps to remove an attribute that is not stored in a property extension database:
   A. Open the `wimxmlextenstaion.xml` file.
   B. Locate and delete the `propertySchema` definition for the attributes you previously added:

   ```xml
   <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String" multiValued="true" propertyName="attribute_name">
     <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
   </wim:propertySchema>
   ```
   C. Save your changes to the `wimxmlextenstaion.xml` file.
   D. Open the `wimconfig.xml` file.
   E. Locate and delete the stanza that corresponds to the custom attribute you deleted from the `wimextension.xml` file; for example:

   ```xml
   <config:attributes name="attribute_name" propertyName="property_name">
     <config:entityTypes>PersonAccount</config:entityTypes>
   </config:attributes>
   ```
   F. Save your changes to the `wimconfig.xml` file.
G. Stop and restart the server1 and WebSphere_Portal servers.

Parent topic: Adapting the attribute configuration
Previous topic: Mapping attributes on IBM i5/OS in a clustered environment
Preparing additional nodes on IBM i5/OS

After installing and configuring your primary node, you can create your secondary nodes. You must install IBM® WebSphere® Portal on each node and then configure the node to access the database and user registry before adding it to the cluster.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on IBM i5/OS
  - Preparing your IBM i5/OS
  - Preparing the primary node on IBM i5/OS
  - Installing a secondary node after migrating from previous version

Perform the following tasks to prepare your secondary node:

1. **Installing WebSphere Portal on i5/OS on the additional nodes**
   Install IBM WebSphere Portal on your secondary nodes to create a highly available and scalable environment.

2. **Adding additional nodes to the cluster on i5/OS**
   After installing IBM WebSphere Portal on your secondary node, you must add the node to the cluster to create a highly available environment.

3. **Optional: Adding vertical cluster members to a static cluster on IBM i5/OS**
   You can add vertical cluster members to share the workload demands of your cluster across multiple members running on the same physical machine.

**Parent topic:** Setting up a cluster on IBM i5/OS

**Previous topic:** Configuring WebSphere Portal to use a user registry on IBM i5/OS in a clustered environment

**Next topic:** Rendering documents on IBM i5/OS
Installing WebSphere Portal on i5/OS on the additional nodes

Install IBM® WebSphere® Portal on your secondary nodes to create a highly available and scalable environment.

Review the Supported hardware and software requirements for your operating system and ensure that all hardware and software matches the minimum product level before installing WebSphere Portal. Review Installation methods, options, and sources before installing to determine your installation source location and installation method. Note: If you plan to use IBM WebSphere Application Server Version 7.0, you must purchase and install it before installing WebSphere Portal on the existing WebSphere Application Server Version 7.0 instance.

Perform the following steps to install WebSphere Portal:

1. Type ping yourserver.yourcompany.com on a command line to verify that your fully qualified host name is properly configured.
2. Setup a static IP address on the server where you will install WebSphere Portal.
3. If you are installing with an existing WebSphere Application Server instance, ensure it is installed at the supported level and has the following features installed:
   - Restriction: Installing WebSphere Portal Version 6.1.0.1 or higher on an existing WebSphere Application Server Version 7.0 instance where WebSphere Portal Version 7.0 was installed is unsupported.
   - Important: Besides ensuring that the existing WebSphere Application Server instance is at the supported level, you will also need to ensure that Apache Derby is installed at the supported level before installing WebSphere Portal. See WebSphere Portal detailed system requirements for supported levels.

   - Application Server
   - Administration
     - Scripted Administration
     - Administrative Console
   - Ant and Deployment Tools
     - Deploy Tool
     - Ant Utilities

4. Choose one of the following installation commands based on the installation method you want to use to install WebSphere Portal; see Installation Methods for more information:

   **Advance installation parameters:** To customize your installation, you can add various parameters to your installation commands; for example, you can specify your port information. See Advanced installation parameters for information.

   **Restriction:** When defining your installation path, the ONLY supported characters are ASCII letters [A-Z and a-z], numbers [0-9], slashes [/ or \, depending on your operating system], and the dash [-].

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphical user Interface</td>
<td>install400.bat Optional attribute: WebSphere Application Server profiles and configurations are performed with the Classic 64-bit JVM by default. To install and configure the portal profile with J9 32-bit JVM, add the -W enableClassicJVM.active=false attribute to your installation command.</td>
</tr>
</tbody>
</table>
5. Perform the following steps if you migrated from a previous version of WebSphere Portal:

A. Log on to the WebSphere Application Server Administrative Console.


C. Click WP Identification.

D. Click Custom properties.

E. If applicable, note the location of the LpidToGupidMapping.properties file.

F. If the LpidToGupidMapping.properties file exists on the primary node, copy the file to the same location on the secondary node.

Parent topic: Preparing additional nodes on IBM i5/OS

Next topic: Adding additional nodes to the cluster on i5/OS

Related concepts

Installation methods
IBM Support Assistant Lite for WebSphere Portal
Adding additional nodes to the cluster on i5/OS

After installing IBM® WebSphere® Portal on your secondary node, you must add the node to the cluster to create a highly available environment.

- Prerequisites
  - Installing WebSphere Portal on i5/OS on the additional nodes

Perform the following steps to add the secondary node to the cluster:

1. Perform the following steps on the secondary node to access your database server:
   - Note: All properties files are located in the \wp_profile_root\ConfigEngine\properties directory.
     A. For JDBC Type 2 connections only, ensure that the database client software is installed on the secondary node using the same settings as the primary node and that you can connect to the remote database. It is not necessary to create databases or users when configuring secondary nodes.
     B. For JDBC Type 4 connections only, copy the JDBC Type 4 jar files to the secondary node from the DB2 server.
     C. Set all database properties in the wkplc_dbtype.properties and wkplc_comp.properties files with the appropriate values from the primary node.

2. Run the following tasks to validate your database settings:
   A. ConfigEngine.sh validate-database-driver
      DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   B. ConfigEngine.sh validate-database-connection -DWasPassword=password
      DTransferDomainList=release,customization,community,jcr,feedback,likeminds

3. Stop the server1 and WebSphere_Portal servers on the secondary node and ensure that the following parameters are set correctly in the wkplc.properties file, located in the profiles/wp_profile/ConfigEngine/properties directory of the secondary node:
   - Note: Although you can add these parameters directly to any task that you run while creating your cluster, you may want to add them to the properties file while creating your cluster and then remove them when you are finished to keep your environment secure.
     A. Verify that WasUserid is set to your deployment manager administrator ID.
     B. Verify that WasPassword is set to your deployment manager administrator password.
     C. Verify that PortalAdminId is set to your WebSphere Portal administrator ID.
     D. Verify that PortalAdminPwd is set to your WebSphere Portal password.
     E. Verify that WasRemoteHostName is set to the fully qualified host name of the deployment manager.
     F. Verify that WasSoapPort is set to the SOAP port that the deployment manager is using; the default value is 8879.
     G. Verify that PrimaryNode is set to false.
     H. Verify that ClusterName is set to the primary node’s ClusterName.

4. If the Deployment Manager is configured to use a stand-alone LDAP user registry, update the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory, on the secondary node with the stand-alone LDAP user registry property values from the Deployment Manager. You can find these settings under the VMM Stand-alone LDAP configuration heading.

5. Run the ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=dmgr_password task, from the wp_profile_root/ConfigEngine directory of the secondary node.

6. Optional: If you configured a database user registry or a property extension (lookaside) database, run the following task to set up access to the database drivers:
**Note:** You will also need to run this task if you migrated the primary node from a release prior to Version 6.1.0, even if you did not configure a database user registry or a property extension (lookaside) database.

A. Set the property value for `federated.db.DbType` if using a database user registry or set the property value for `la.DbType` if using a property extension database in the `wkplc.properties` file.

  **Note:** If you migrated the primary node from a version prior to Version 6.1.0 and you are not using a database user registry or a property extension database, set the property value for `federated.db.DbType` to the same value that is in the `wkplc.properties` file on the primary node.

B. Run the `ConfigEngine.sh wp-node-prep-vmm-db-secured-environment` task from the `wp_profile_root/ConfigEngine` directory to create the variable used to access the VMM database jars. **Note:**`VmmNodeName` is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The `db_type` in `db_type.NodeDbLibrary` should be set to the type of database you are using, for example `db2`.

```
DB2 for i5/OS Type 2 driver:/QIBM/ProdData/Java400/ext/db2_classes.jar

DB2 for i5/OS Type 4 driver:/QIBM/ProdData/HTTP/Public/jt400/lib/jt400.jar
```

7. Run the `ConfigEngine.sh cluster-node-config-post-federation` task. **Note:** If you want to customize the WebSphere Portal server name (WebSphere_Portal) for your additional node, set the `ServerName` parameter in the `wkplc.properties` file after running the `cluster-node-config-post-federation` task.

8. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal Program ID and password to match an administrative user defined in the cell’s user registry. Run the `ConfigEngine.sh wp-change-portal-admin-user` task from the `wp_profile_root/ConfigEngine` directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

  **Attention:** The password value for `-DWasPassword` is the Deployment Manager administrative password.

  **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

  **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

  **Fast path:** If the value for `newAdminGroupId` contains a space; for example `Software Group`, open the `wkplc.properties` file and add the values for `newAdminId`, `newAdminPw`, and `newAdminGroupId`. Save your changes and run the `ConfigEngine.bat wp-change-portal-admin-user` task.

  **Important:** If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user` task until after the `cluster-node-config-cluster-setup` (static cluster) or `cluster-node-config-dynamic-cluster-setup` (dynamic cluster) task completes. After running the `wp-change-portal-admin-user` task, start or restart the WebSphere_Portal server to use the updated administrator user ID.

  **Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the `wp-change-portal-admin-user` task. `-DnewAdminPw` is an optional parameter to update the Administrative password in the `wkplc.properties` file if required.

9. Optional: Open the `wkplc.properties` file and change the `ServerName` parameter from the default `WebSphere_Portal_nodename` value to a value that meets your business needs. Do not change the parameter to `WebSphere_Portal` as this is the primary node value.

10. Run the `ConfigEngine.sh cluster-node-config-cluster-setup` task to add the node to your cluster.
11. Wait at least 30 minutes before stopping and restarting the deployment manager because the node agent runs a background thread to expand all EAR files into their target directories. **Note:** Attempting to restart the deployment manager while the expansion is in process can result in an unusable application. Although the time to complete an EAR expansion is environment-dependent, a 30 minute wait is sufficient for a typical environment.

12. Perform the following steps to access the Web Content Management content through an external Web server:
   A. Log on to the deployment manager administrative console.
   B. Select **Environment > WebSphere Variables**.
   C. From the **Scope** drop-down menu, select the **Node=nodename, Server=servername** option to narrow the scope of the listed variables, where **Node=nodename** is the node that contains the application server.
   D. Update the **WCM_HOST** variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.
   E. Update the **WCM_PORT** variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.
   F. Perform the following steps to synchronize the node with the deployment manager:
      1. Select **System Administration > Nodes**.
      2. Select the node that you want to synchronize from the list.
      3. Click **Full Resynchronize**.
   G. Log off of the deployment manager administrative console.

13. If you entered passwords in any of the properties files while creating your cluster, you should remove them for security purposes. See "Deleting passwords from properties files" under Related tasks for information.

**Parent topic:** Preparing additional nodes on IBM i5/OS  
**Previous topic:** Installing WebSphere Portal on i5/OS on the additional nodes  
**Next topic:** Adding vertical cluster members to a static cluster on IBM i5/OS

**Related tasks**
Deleting passwords from properties files
Adding vertical cluster members to a static cluster on IBM i5/OS

You can add vertical cluster members to share the workload demands of your cluster across multiple members running on the same physical machine.

- **Prerequisites**
  - Installing WebSphere Portal on i5/OS on the additional nodes
  - Adding additional nodes to the cluster on i5/OS

Perform the following steps to add a vertical cluster member to your clustered environment:

1. Log into the deployment manager administrative console.
2. Depending on your version of WebSphere Application Server, make the appropriate selection from the navigation:
   - For **WebSphere Application Server Version 6.1**: Click **Servers > Clusters** in the console navigation tree, select the cluster name, and then click **Cluster members** from the list of additional properties.
   - For **WebSphere Application Server Version 7.0**: Click **Servers > Clusters > WebSphere application server clusters > cluster_name > Cluster members**.
3. Click **New** to create the cluster member.
   - Define the name of cluster member. **Note:** Do not use spaces in the cluster member name.
   - Select an existing node where IBM® WebSphere® Portal is installed.
   - Check the box **Generate Unique HTTP Ports**.
   - Click **Apply** and then click **Next** to view the summary.

**Important:** If you are not using an external Web server in your clustered environment, you must update the virtual host entries for the new port created when adding a cluster member. You can do this by updating the `default_host` virtual host in the administrative console and adding a new alias entry for the port number (use an "*" wildcard character for the host name). See **Configuring virtual hosts** for information.

4. Click **Finish**, and save the changes.
   - The new cluster topology can be viewed from the **Servers > Cluster Topology** view. If using WebSphere Application Server Version 7.0, the new cluster topology can be viewed from the **Servers > Clusters > Cluster Topology** view.
   - The **Servers > Application Servers** view will list the new server cluster members. If using WebSphere Application Server Version 7.0, the **Servers > Server Types > WebSphere application servers** view will list the new server cluster members.

5. Perform the following steps to enable cache replication:
   - From the deployment manager Administrative Console, navigate to **Servers > Application servers** and then click the new vertical cluster member(s). If using WebSphere Application Server Version 7.0, navigate to **Servers > Server Types > WebSphere application servers** and then click the new vertical cluster member(s).
   - Click **Dynamic cache service** under **Container services**.
   - Change **Cache size** to **3000** entries.
   - Check the **Enable cache replication** check box.
   - Select **NOT_SHARED** from the **Replication type** drop-down menu.
   - Click **OK**.
   - Click **Save** to save your changes to the master configuration.

6. Perform the following steps on each vertical cluster member to clean up the server-scoped resources, caches, and resource providers:
   - Run the `ConfigEngine.sh cluster-node-config-vertical-cluster-setup -DServerName=unique vertical`
Perform the following steps to access the Web Content Management content through an external Web server:

A. Log on to the deployment manager administrative console.

B. Select **Environment > WebSphere Variables**.

C. From the **Scope** drop-down menu, select the **Node=nodename, Server=servername** option to narrow the scope of the listed variables, where **Node=nodename** is the node that contains the application server.

D. Update the **WCM_HOST** variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.

E. Update the **WCM_PORT** variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.

F. Perform the following steps to synchronize the node with the deployment manager:
   1. Select **System Administration > Nodes**.
   2. Select the node that you want to synchronize from the list.
   3. Click **Full Resynchronize**.

G. Log off of the deployment manager administrative console.

8. Save your changes and resynchronize the nodes.
   A. In the administrative console for the deployment manager, click **Save** on the task bar, and save your administrative configuration.
   B. Select **System Administration > Nodes**, select the node from the list, and click **Full Resynchronize**.

9. Regenerate the Web server plug-in.
   A. Regenerate the Web server plug-in using the deployment manager administrative console.
   B. If you are using a remote Web server, copy the updated plug-in configuration file (**plugin-cfg.xml**) to the Web server's plug-in configuration directory.

10. Stop and start the Web server.
Rendering documents on IBM i5/OS

In order to enable document preview functionality for IBM® Lotus Web Content Management and the Common Mail portlet, you must set up an HTML rendering server to work with WebSphere Portal. Because IBM i5/OS does not contain native graphics support, you must install additional fonts to perform the document conversion required by these functions. Document conversion enables WebSphere Portal to convert documents produced by commonly used office programs into Web pages, so that they can be viewed and searched by users online. The additional fonts include an HTML rendering server known as X virtual frame buffer for the X server.

Before you begin, you must have:
- A WebSphere Portal profile running on your IBM i5/OS system; record the name of your WebSphere Portal profile for future reference.
- OS/400 - Additional Fonts (5722SS1, Option 43) or OS/400 - Additional Fonts (5761SS1, Option 43) installed.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on IBM i5/OS
  - Preparing your IBM i5/OS
  - Preparing the primary node on IBM i5/OS
  - Preparing additional nodes on IBM i5/OS

Perform the following tasks on each WebSphere Portal node to render documents on IBM i5/OS:

**Note:** The HTML rendering server (X virtual frame buffer for X server) that you associate with your WebSphere Portal profile should only be used for WebSphere Portal. Using the HTML rendering server with other applications may cause problems.

1. Configuring an HTML rendering server on IBM i5/OS
   
   After you have installed OS/400 - Additional Fonts (5722SS1, Option 43), which is for Version 5R4, or OS/400 - Additional Fonts (5761SS1, Option 43), which is for Version 6R1, on your IBM i5/OS system, an HTML rendering server (X virtual frame buffer for X server) is present. You must select a display number for the HTML rendering server.

2. Optional: Associating an HTML rendering server with WebSphere Portal
   
   After selecting a display number for the HTML rendering server (X virtual frame buffer for X server), you must associate this server with the installed IBM WebSphere Portal profile.
Configuring an HTML rendering server on IBM i5/OS

After you have installed OS/400 - Additional Fonts (5722SS1, Option 43), which is for Version 5R4, or OS/400 - Additional Fonts (5761SS1, Option 43), which is for Version 6R1, on your IBM® i5/OS system, an HTML rendering server (X virtual frame buffer for X server) is present. You must select a display number for the HTML rendering server.

Perform the following steps to configure an HTML rendering server:

1. If the QShell Interpreter is running, perform the following commands on the IBM i5/OS command line to stop it:
   - QSH
   - Press F3

2. Type `CALL QP2TERM` on the command line to start the OS/400 Portable Application Solutions Environment (OS/400 PASE) console.

3. Type `ps gaxuw | grep Xvnc ; ps gaxuw | grep vfb` on the command line to list all active HTML rendering servers:
   
   **Note:** If other rendering servers are already active, you may see output such as this (the number following the colon is the display number already in use):
   
   ```
   v2kea554 40571 0.0 0.0 12484 0 - A Jul 13 4:08
   /QOpenSys/QIBM/ProdData/DeveloperTools/vnc/Xvnc :6 -desktop X -httpd
   ```

4. Select any number from 1 to 99 that is not in use.

5. Type `SBMJOB CMD(CALL PGM(QP2SHELL) PARM(`/usr/bin/X11/X' '-vfb' '':N')) JOB(XVFB) JOBQ(QSYSNOMAX) ALWMLTTHD(*YES)` on the command line to start the HTML rendering server, where \(N\) is the display number.

6. Verify that the HTML rendering server is started by repeating the prior steps to start PASE and list the active servers, confirming that an HTML rendering server with your display number is in the list.

**Parent topic:** Rendering documents on IBM i5/OS

**Next topic:** Associating an HTML rendering server with WebSphere Portal
Associating an HTML rendering server with WebSphere Portal

After selecting a display number for the HTML rendering server (X virtual frame buffer for X server), you must associate this server with the installed IBM® WebSphere® Portal profile.

- Prerequisites
  - Configuring an HTML rendering server on IBM i5/OS

Perform the following steps to associate the HTML rendering server (X virtual frame buffer for X server) with WebSphere Portal:

1. Log into the IBM WebSphere Application Server administrative console.
2. Log into the IBM WebSphere Application Server administrative console. Depending on your version of WebSphere Application Server, make the appropriate selection from the navigation:
   - Click New.
   - In the Name field type DISPLAY.
   - In the Value field type host_name:n, where host_name is the TCP/IP host name of your system and n is the display number of the HTML rendering server. (Example: 'mysystem.xland.company.com:1').
3. Click OK.
4. Save your changes to the master WebSphere Application Server configuration file.
5. Perform the following steps to stop and restart the server1 and WebSphere_Portal servers, where server1 is the name of the WebSphere Application Server and WebSphere_Portal is the name of the WebSphere Portal server:
   A. Open a command prompt and change to the wp_profile_root/bin directory.
   B. Enter the stopServer server1 -username admin_userid -password admin_password command to stop the WebSphere Application Server.
   C. Enter the stopServer WebSphere_Portal -username admin_userid -password admin_password command to stop the WebSphere Portal server.
   D. Enter the startServer server1 command.
   E. Enter the startServer WebSphere_Portal command.
Tune your servers

Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on IBM i5/OS
  - Preparing your IBM i5/OS
  - Preparing the primary node on IBM i5/OS
  - Preparing additional nodes on IBM i5/OS

**Base Portal Tuning Scenarios**

Tuning a WebSphere Portal environment involves tuning and configuring the various systems and components of the environment. The tuning guide provides general concepts and details specifics configuration instructions. Instructions are included for:

- Configuring the application server and the resources defined for that application server
- Determining the cloning strategy for expanding or extending the environment
- Tuning the database(s) and database server
- Tuning the directory server and its database
- Tuning the Web server
- Tuning the operating system and network
- Tuning the WebSphere Portal services

**Parent topic:** Setting up a cluster on IBM i5/OS  
**Previous topic:** Rendering documents on IBM i5/OS  
**Next topic:** Configuring search in a cluster on i5/OS

**Related information**

- WebSphere Portal and Lotus Web Content Management 6.1.x Performance Tuning Guide
- Tuning a Cluster Environment
- IBM WebSphere Portal Performance Troubleshooting Guide
Configuring search in a cluster on i5/OS

IBM® WebSphere® Portal provides two distinct search capabilities. You can utilize both types of search capabilities in a clustered environment.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on IBM i5/OS
  - Preparing your IBM i5/OS
  - Preparing the primary node on IBM i5/OS
  - Preparing additional nodes on IBM i5/OS
  - Tune your servers

- **Configuring Portal Search in a cluster on i5/OS**
  To support Portal Search in a clustered environment, you must install and configure search for remote search service on an IBM WebSphere Application Server node that is not part of the IBM WebSphere Portal cluster.

- **Configuring JCR search in a cluster on i5/OS**
  To enable search in a cluster for content stored in the JCR database, you must configure each server in the cluster to access a shared directory. JCR-based content includes content created with Web Content Management or Personalization.

Parent topic: Setting up a cluster on IBM i5/OS
Previous topic: Tune your servers
Next topic: Setting up multiple clusters on IBM i5/OS
Configuring Portal Search in a cluster on i5/OS

To support Portal Search in a clustered environment, you must install and configure search for remote search service on an IBM® WebSphere® Application Server node that is not part of the IBM WebSphere Portal cluster.

To install and configure the search service remotely, perform the following tasks:

1. Install and configure the search service to work remotely, that is, on a remote WebSphere Application Server node which is not part of the portal cluster. You can provide the remote search service either as an EJB or as a Web service via SOAP. Deploy the appropriate EJB or SOAP EAR file on the remote WebSphere Application Server node. For details about how to do this, refer to the WebSphere Application Server documentation.

2. Configure the search portlets for remote search service so that they access the remote server accordingly.

**Notes:**

1. If you have configured a remote search service for a portal cluster, you need to configure the default location for search collections to a directory on the remote server that has write access.

2. The portal site default search collection is created only once at the first time when an administrator selects the search administration portlet Manage Search. If this occurred _before_ you configure the portlet for remote search, then the default portal site search collection is only available on the primary node of the cluster, but not on the remote server. In this case you need to recreate the portal site collection to make it available for search on all nodes of the cluster.

**Parent topic:** Configuring search in a cluster on i5/OS

**Related concepts**

Planning and preparing for Portal Search

Using remote search service

**Related tasks**

Configuring Portal Search for remote search service

Configuring the Search and Browse portlet for remote search service

Configuring the default location for search collections

Creating or resetting the portal site collection

Configuring the primary node to communicate with the deployment manager on i5/OS

Configuring a remote search service
Configuring JCR search in a cluster on i5/OS

To enable search in a cluster for content stored in the JCR database, you must configure each server in the cluster to access a shared directory. JCR-based content includes content created with Web Content Management or Personalization.

Create a shared directory called `jcr/search` on a server in the network and ensure that each node in the cluster and the Deployment Manager has network access to the directory.

**Note:** If you are creating content in a clustered environment using the authoring portlet provided with Web Content Management, additional configuration steps are required to enable content created by these content features to be searchable in a cluster.

Perform the following steps on each server in the cluster to configure Search in a clustered environment:

1. Edit the `icm.properties` file, located in the `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm` directory.
2. Change the value of the `jcr.textsearch.indexdirectory` property to the shared directory; for example,
   
   `jcr.textsearch.indexdirectory=\\your_server\your_share\jcr\search`

   You can specify the shared directory value in one of the following formats: *Table 1. The format for the shared directory.*

<table>
<thead>
<tr>
<th>Format</th>
<th>Shared directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal Naming Convention (UNC) format</td>
<td><code>\\your_server\your_share\jcr\search</code> Example: <code>\\hostname.example.com\share\jcr\search</code></td>
</tr>
<tr>
<td>Mounted resource format (with forward slashes)</td>
<td><code>/your_share/jcr/search</code> For example: <code>/mnt/jcr/search</code></td>
</tr>
</tbody>
</table>

**Important:** This format requires that you mount the shared directory to the local server (for example, through a mapped network drive or a mounted directory). When using the mounted resource format, always use forward slashes instead of back slashes, regardless of the native operating system path format.

3. Required: Perform the following steps to delete the default search collections from the Manage Search portlet:

   A. Log on to WebSphere Portal as an administrator.
   B. Click **Administration > Search Administration > Manage Search**.
   C. Click **Search Collections**.
   D. Click the **Delete Collection** icon for the **Portal Content** search collection.
   E. Click **OK**.
   F. Restart the WebSphere_Portal server.
   G. Go to the Manage Search portlet and confirm that the **Portal Content** search collection was deleted.

**Parent topic:** Configuring search in a cluster on i5/OS
Setting up multiple clusters on IBM i5/OS

The majority of the steps to build another cluster are the same as when building the first cluster in the same cell, with a few exceptions. Basically, the new profile will be designated as the primary profile, using IBM® WebSphere® Portal clustering terminology, and will then be used as the basis for the new cluster definition. This duplicates the build process of the first cluster in the cell. During the federation process, if any applications on this new node already exist in the cell (because they are in use by the first cluster), then Deployment Manager will not allow them to be added. After federation, the applications that already exist in the cell are not mapped to the WebSphere_Portal server on the newly federated node, and thus the existing applications must be remapped to this newly federated server to restore its application list. Therefore, depending on the configuration of the new profile, there will likely be some combination of applications shared with the other existing cluster, and some applications unique to this new profile.

Before attempting alternative approaches for building multiple portal-based clusters within a single cell, please contact IBM.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on IBM i5/OS
  - Preparing your IBM i5/OS
  - Preparing the primary node on IBM i5/OS
  - Preparing additional nodes on IBM i5/OS
  - Tune your servers

- **Installing multiple clusters in a single cell on IBM i5/OS**
  Create a new, independent IBM WebSphere Portal cluster in a cell where a WebSphere Portal cluster already exists.

- **Routing requests across clusters on IBM i5/OS**
  The HTTP Server plug-in that comes with IBM WebSphere Application Server is typically used to balance requests for applications across members of the cluster. While an application can be mapped to more than one cluster, automatic plug-in generation does not provide routing or balancing traffic for the same application across multiple clusters. Multiple cluster environments with shared applications therefore cannot rely solely on WebSphere Application Server automatic plug-in generation to be able to route requests using a web server. One option in this scenario is developing a customized method for defining and maintaining the plug-in configuration file used by the Web server to provide for the required routing of user requests.
Installing multiple clusters in a single cell on IBM i5/OS

Create a new, independent IBM® WebSphere® Portal cluster in a cell where a WebSphere Portal cluster already exists. In the following steps, Cluster A will be used to describe the existing cluster. Portal B will be used to describe the new server profile that will be the basis for the new cluster definition, Cluster B. Perform the following steps to install multiple clusters in a single cell:

1. Upgrade Cluster A, including the Deployment Manager node, to the current, supported hardware and software levels and to the current version of IBM WebSphere Portal.

2. Install and configure Portal B; see the "Preparing the primary node" topic for the appropriate operating system for details. Important: Maintain the same number of data sources with identical names to the Cluster A data sources so that data source bindings in the applications can be resolved on every cluster in which they run. If implementing database sharing across the clusters, the above statement refers to both the shared and non-shared domains; all domains should use the same names.

3. Optional: Using the same database user ID and password for each identically named domain/data source will allow the existing JAAS AuthenticationAliases to be functional. If unique database user ID and password are required, additional manual configuration is needed to create new JAAS AuthenticationAliases for each data source and map these accordingly. On the primary node of Cluster A, run the ConfigEngine.sh create-alias-multiple-cluster -DauthDomainList=release,jcr -DWasPassword=dmgr_password task from the wp_profile_root/ConfigEngine directory to create the new JAAS AuthenticationAliases. Where authDomainList is set to a list of domains which use unique database user ID and passwords and those domain properties are set correctly in the wkplc_comp.properties file, including user ID and password.

4. Optional: If necessary, upgrade Portal B to the current cumulative fix.

5. Run the ConfigEngine.sh mapped-app-list-create -DWasPassword=password task from the wp_profile_root/ConfigEngine directory to build an inventory of Portal B enterprise applications and portlets.

6. Stop the server1 and WebSphere_Portal servers on Portal B and ensure that the following parameters are set correctly in the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory:
   - Set WasSoapPort to the deployment manager's port.
   - Set WasRemoteHostName to the full host name of the deployment manager.

7. If the Deployment Manager is configured to use a stand-alone LDAP user registry, update the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory, on the primary node with the stand-alone LDAP user registry property values from the Deployment Manager. You can find these settings under the VMM Stand-alone LDAP configuration heading. Important: Ensure that you set WasUserId and WasPassword to the Deployment Manager user ID and password.

8. Run the ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=password task from the wp_profile_root/ConfigEngine directory of the primary node.

9. Run the ConfigEngine.sh map-apps-to-server -DWasPassword=password task to determine which applications from the inventory list are no longer mapped to Portal B. The task uses the application profiles already in the cell to restore the mappings. Wait 30 minutes after running this task to allow all EAR files to expand before proceeding to the next step.

10. Perform the following steps to federate Portal B into the deployment manager cell:
A. Ensure that all database parameters are correctly set, including passwords, in the `wkplc_comp.properties` and `wkplc_dbtype.properties` files.

B. Run the `ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=dmgr_password` task.

C. After running the `cluster-node-config-post-federation` task, wait at least 30 minutes to allow all EAR files to expand.

D. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell's user registry. Run the `ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task, from the `wp_profile_root/ConfigEngine` directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

**Important:** If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user` task until after the `cluster-node-config-cluster-setup` (static cluster) or `cluster-node-config-dynamic-cluster-setup` (dynamic cluster) task completes. After running the `wp-change-portal-admin-user` task, start or restart the WebSphere _Portal_ server to use the updated administrator user ID.

**Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the `wp-change-portal-admin-user` task. `-DnewAdminPw` is an optional parameter to update the Administrative password in the `wkplc.properties` file if required.

E. From the administrative console, click *System Administration > Node Agents.*

F. Check the box next to the required node agent and then click Restart.

G. Stop and restart the deployment manager.

H. Stop and restart the WebSphere _Portal_ server on Portal B

11. Restart the WebSphere _Portal_ server on Cluster A. Verify that Cluster A is functionally intact by spot checking pages and portlets and then verify that Portal B is functionally intact by spot checking pages and portlets that you deployed into Portal B before it was federated. Any discrepancies or errors should be corrected now before continuing. **Note:** If Portal B is using a non-default Portal server administrative ID, not *wpsadmin*, the server will not be functional until the cluster configuration is complete and the Portal administrative ID has been configured to match the Cells security settings.

12. Perform the following steps to define a cluster using Portal B as the basis:

A. Run the `ConfigEngine.sh cluster-node-config-cluster-setup -DWasPassword=dmgr_password` task.

B. Configure the cluster to use an external Web server to take advantage of features such as workload management. Choose one of the following options:
   - Recommended remote distributed installation
   - Local distributed installation
   - Setting up multiple clusters on IBM i5/OS

C. Perform the following steps to access the Web Content Management content through an external Web server:
   1. Log on to the deployment manager administrative console.
   2. Select Environment > WebSphere Variables.
   3. From the Scope drop-down menu, select the Node=node_name, Server=servername option to narrow the scope of the listed variables, where Node=node_name is the node that contains the application server.
   4. Update the WCM_HOST variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.
   5. Update the WCM_PORT variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.

D. Save your changes and then restart the deployment manager, the node agent(s), server1, and the WebSphere _Portal_ servers.
E. Change the task list settings to point to the cluster instead of a single server if using Portal business process integration. See one of the following sections in the Information Center for information:
- Cross cell: Adding a BPI-enabled portal server to a managed cell in a cross-cell setup
- Single cell: Adding a BPI-enabled portal server to a managed cell in a single-cell setup

13. Install any additional nodes to the cell to support additional cluster members for Cluster B identically to the primary node, and then federate as them as secondary nodes and define as cluster members on these nodes. For information about adding additional nodes navigate to Installing WebSphere Portal > Setting up WebSphere Portal > Setting up a clustered production environment > Setting up a production environment on i5/OS > Preparing additional nodes on i5/OS. You can add additional nodes and/or vertical clusters.

14. Restart the server1 and WebSphere_Portal servers on Cluster A and Cluster B.

15. If you entered passwords in any of the properties files while creating your cluster, you should remove them for security purposes. See “Deleting passwords from properties files” under Related tasks for information.

Installation of Cluster B is complete. It is now an independent cluster from Cluster A, which means that Cluster B can have its own configuration, set of end-user portlets, and target community. Any applications that are common between Cluster A and Cluster B are most likely infrastructure or related to administration, and special care needs to be taken to preserve their commonality between clusters and correct maintenance levels.

Parent topic: Setting up multiple clusters on IBM i5/OS

Related tasks
Deleting passwords from properties files
Adding a BPI-enabled portal server to a managed cell in a cross-cell setup
Adding a BPI-enabled portal server to a managed cell in a single-cell setup
Routing requests across clusters on IBM i5/OS

The HTTP Server plug-in that comes with IBM® WebSphere® Application Server is typically used to balance requests for applications across members of the cluster. While an application can be mapped to more than one cluster, automatic plug-in generation does not provide routing or balancing traffic for the same application across multiple clusters. Multiple cluster environments with shared applications therefore cannot rely solely on WebSphere Application Server automatic plug-in generation to be able to route requests using a web server. One option in this scenario is developing a customized method for defining and maintaining the plug-in configuration file used by the Web server to provide for the required routing of user requests.

An important consideration in a multiple cluster environment is ensuring that all subsequent HTTP requests for an end user are routed to the same cluster that processed the first HTTP request. The WebSphere Portal login processing depends upon preserving this cluster affinity during this initial time until the user has successfully logged in and session cookies maintain affinity. In order to guarantee that affinity is preserved during login, set the Navigator Service `public.session` parameter to a value of true. Refer to “Portal Configuration Services” for information on how to configure this parameter.

Parent topic: Setting up multiple clusters on IBM i5/OS
Sharing database domains between clusters on i5/OS

To help provide redundancy and failover support in production environments composed of multiple clusters in a single cell and multiple clusters in different cells, you can share database domains between those clusters. IBM® WebSphere® Portal data is organized into different database domains, with different availability requirements depending on how the production environment is set up. When multiple lines of production are involved and each line of production is implemented as a cluster of servers, sharing database domains ensures that the data is automatically synchronized across the lines of production.

- Prerequisites
  - Preparing prerequisite and corequisite software on IBM i5/OS
  - Preparing your IBM i5/OS
  - Preparing the primary node on IBM i5/OS
  - Preparing additional nodes on IBM i5/OS
  - Tune your servers

Important: JCR domains and release domains cannot be shared between different Web Content Management clusters or servers. Each distinct cluster or server in your Web Content Management system must use a separate JCR or release domain. For example:

<table>
<thead>
<tr>
<th>Development Server</th>
<th>Authoring Cluster</th>
<th>Staging Server</th>
<th>Delivery Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCR Domain 1</td>
<td>JCR Domain 2</td>
<td>JCR Domain 3</td>
<td>JCR Domain 4</td>
</tr>
<tr>
<td>Release Domain 1</td>
<td>Release Domain 2</td>
<td>Release Domain 3</td>
<td>Release Domain 4</td>
</tr>
</tbody>
</table>

Perform the following steps to share database domains when setting up an environment with multiple clusters:

1. Set up the first cluster (referred to as Cluster A in these instructions).
2. Determine which database domains you want to share with any other clusters in the environment.
3. Install the primary node of the next cluster (Cluster B), and perform the following steps to configure the node to use the shared database domains.
   A. Perform a partial database transfer of the database domains that you are not sharing. For example, if you are sharing only the Customization and Community domains, you would transfer the remaining domains to the database you are using for Cluster B.
   B. Reconfigure the shared database domains on the node to connect to the shared database domains you are using for Cluster A. For example, to connect to the Customization and Community domains for Cluster A, you would connect to those domains from Cluster B.
4. Continue setting up the primary node as described in the cluster instructions.
5. Install the secondary node of Cluster B, and perform the following steps to configure the node to use the shared database domains.
   A. For those database domains that you are not sharing between clusters, reconfigure the domains to connect to the database domains you are using for Cluster B. As in the example for the primary node, if you are sharing only the Customization and Community domains, reconfigure the remaining domains on the secondary node to use the domains of the primary node for Cluster B.
B. Reconfigure the shared database domains on the node to connect to the shared database domains you are using for Cluster A. For example, to connect to the Customization and Community domains for Cluster A, you would connect to those domains from Cluster B.

6. Continue setting up the secondary node as described in the cluster instructions.

**Parent topic:** Setting up a cluster on IBM i5/OS

**Previous topic:** Setting up multiple clusters on IBM i5/OS

**Related tasks**

Connecting to existing database domains
Setting up a cluster on Linux

In a production environment, you can install and configure IBM® WebSphere® Portal on a single (primary) node and then add additional nodes in the future as your company's needs grow. You can also deploy a high availability production environment by creating and configuring clustered deployments of multiple servers running WebSphere Portal managed by IBM WebSphere Application Server Network Deployment.

- **Prerequisites**
  - Technotes for installation and configuration issues

Perform the following steps to set up your production environment on Linux:

1. **Preparing prerequisite and corequisite software on Linux**
   Before creating your high availability production environment, you must prepare your database, your user registry, your deployment manager, and your remote Web server.

2. **Preparing your Linux operating system**
   View information on setting up your operating system for IBM WebSphere Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

3. **Preparing the primary node on Linux**
   Before creating your high availability environment, you must install IBM WebSphere Portal on your primary node and then configure your database and your network deployment manager.

4. **Choosing the type of cluster to create on Linux**
   If you installed a IBM WebSphere Application Server Network Deployment, create a static cluster to handle failover requests. If you installed a IBM WebSphere Virtual Enterprise, create a dynamic cluster to balance member workloads.

5. **Preparing a remote Web server when portal is installed on Linux**
   Install and configure the Web server plug-in, provided by IBM WebSphere Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

6. **Configuring WebSphere Portal to use a user registry on Linux in a clustered environment**
   Configure user registry security on IBM WebSphere Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

7. **Preparing additional nodes on Linux**
   After installing and configuring your primary node, you can create your secondary nodes. You must install IBM WebSphere Portal on each node and then configure the node to access the database and user registry before adding it to the cluster.

8. **Tune your servers**
   Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

9. **Configuring search in a cluster on Linux**
   IBM WebSphere Portal provides two distinct search capabilities. You can utilize both types of search capabilities in a clustered environment.
10. **Setting up multiple clusters on Linux**

   The majority of the steps to build another cluster are the same as when building the first cluster in the same cell, with a few exceptions. Basically, the new profile will be designated as the primary profile, using IBM WebSphere Portal clustering terminology, and will then be used as the basis for the new cluster definition. This duplicates the build process of the first cluster in the cell. During the federation process, if any applications on this new node already exist in the cell (because they are in use by the first cluster), then Deployment Manager will not allow them to be added. After federation, the applications that already exist in the cell are not mapped to the WebSphere Portal server on the newly federated node, and thus the existing applications must be remapped to this newly federated server to restore its application list. Therefore, depending on the configuration of the new profile, there will likely be some combination of applications shared with the other existing cluster, and some applications unique to this new profile.

11. **Sharing database domains between clusters on Linux**

   To help provide redundancy and failover support in production environments composed of multiple clusters in a single cell and multiple clusters in different cells, you can share database domains between those clusters. IBM WebSphere Portal data is organized into different database domains, with different availability requirements depending on how the production environment is set up. When multiple lines of production are involved and each line of production is implemented as a cluster of servers, sharing database domains ensures that the data is automatically synchronized across the lines of production.

*Parent topic:* Setting up a cluster
Preparing prerequisite and corequisite software on Linux

Before creating your high availability production environment, you must prepare your database, your user registry, your deployment manager, and your remote Web server.

Perform the following tasks to prepare prerequisite and corequisite software:

1. **Preparing the WebSphere Application Server Deployment Manager on Linux**
   IBM WebSphere Portal provides a customized installation package (CIP) that includes IBM WebSphere Application Server Network Deployment and all required maintenance packages. Use the supplied discs to install WebSphere Application Server Network Deployment on a dedicated system.

2. **Preparing a remote Web server when portal is installed on Linux**
   Install and configure the Web server plug-in, provided by IBM WebSphere Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

**Parent topic:** Setting up a cluster on Linux  
**Next topic:** Preparing your Linux operating system
Preparing the WebSphere Application Server Deployment Manager on Linux

IBM® WebSphere® Portal provides a customized installation package (CIP) that includes IBM WebSphere Application Server Network Deployment and all required maintenance packages. Use the supplied discs to install WebSphere Application Server Network Deployment on a dedicated system.

Perform the following steps to prepare the WebSphere Application Server Deployment Manager:

1. Set the file descriptor limit to 10240; for example, ulimit -n 10240.
2. Install WebSphere Application Server Network Deployment or use the CIP to upgrade an existing installation. Run the following command: cd_root/Linux/architecture/ifpackage/WAS/install, where cd_root is the root directory of the disc and architecture is the system's processor architecture.
3. Choose one of the following options to create a default deployment manager profile: Important: While creating the default deployment manager profile, enable administrative security. If you use the Profile Management Tool, check the enable administrative security check box. If you use the manageprofile command, add the enableAdminSecurity=true parameter to the command line.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile Management Tool</td>
<td>See Creating a deployment manager profile for information.</td>
</tr>
<tr>
<td>manageprofile commands</td>
<td>See manageprofile commands for information. Note: If you have a 64-bit environment, only the manageprofiles command is supported when creating profiles.</td>
</tr>
</tbody>
</table>

4. Run the following command to start the deployment manager: ./startManager.sh, from the dmgr_profile_root/bin directory.
5. Use the following URL to launch the network deployment administrative console: http://dmgr_hostname:9060/ibm/console, where dmgr_hostname is the fully qualified host name for the WebSphere Application Server Network Deployment.
6. Log into the deployment manager administrative console.
7. Increase the HTTP connection timeouts for the deployment manager.
   A. Click System Administration > Deployment Manager > Web container transport chains.
   B. Increase the timeout values. For the WCInboundAdmin and WCInboundAdminSecure entries listed in the web container transport chains section, complete the following steps to increase the timeout values:
      1. Click HTTP Inbound Channel.
      2. Change the Read timeout value to 180.
      3. Change the Write timeout value to 180.
      4. Save the configuration changes.
8. Change the timeout request period for the Java Management Extensions (JMX) connector.
   A. Click System administration > Deployment Manager > Administration Services > JMX connectors > SOAPConnector > Custom Properties.
   B. Select the requestTimeout property, and increase the value from 600 to 6000.
   C. Save the configuration changes.
9. Update the maximum Java heap size used by the deployment manager:
   A. Click System administration > Deployment manager > Java and Process Management > Process Definition >
Java Virtual Machine.

B. Update the value in the Maximum Heap Size field. For information about appropriate heap sizes see the documentation for your operating system in the Performance Guides located on the WebSphere Portal and Web Content Management Product Documentation page. **Note:** If using a 32-bit operating system, you will need to set the heap size to a lower size than a 64-bit operating system.

C. Click **OK** and then save your changes.

10. Depending on your version of WebSphere Application Server, make the appropriate selection from the navigation:
   - **Note:** If security is not enabled on your deployment manager, see Enabling security for information before performing this step.
   - For **WebSphere Application Server Version 6.1**: Click Security > Secure administration, applications and infrastructure and select Enable Application Security. Then save the configuration changes.
   - For **WebSphere Application Server Version 7.0**: Click Security > Global security and select Enable Application Security. Then save the configuration changes.

11. Verify that the WebSphere Portal administrative users and administrative group exist in the Deployment Manager cell's user registry. Perform the following steps if you need to create the administrative users and group:
   A. Click **Users and Groups** > **Manage Users**.
   B. Click **Create**.
   C. Type the information for the WebSphere Portal administrative users; for example wpsadmin and wpsbind, and then click **Create**.
   D. Click **Users and Groups** > **Manage Groups**.
   E. Click **Create**.
   F. Type wpsadmins as the name of the WebSphere Portal administrative group and then click **Create**.
   G. Click the group you just created; for example wpsadmins.
   H. Click the **Members** tab.
   I. Click **Add Users**.
   J. Search for the users.
   K. Select the users you want to add to the group.
   L. Click **Add** to add the users to the group.
   M. Click **Close** when you are done adding users to the group.
   N. Log out of the administrative console.


13. Run the following tasks to stop and restart the deployment manager:
   A. `./stopManager.sh -username admin_userid -password admin_password`, from the dmgr_profile_root\bin directory
   B. `./startManager.sh`, from the dmgr_profile_root\bin directory

**Parent topic:** Preparing prerequisite and corequisite software on Linux

**Next topic:** Preparing a remote Web server when portal is installed on Linux
Preparing a remote Web server when portal is installed on Linux

Install and configure the Web server plug-in, provided by IBM® WebSphere® Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

- **Prerequisites**
  - Preparing the WebSphere Application Server Deployment Manager on Linux
  - Preparing prerequisite and corequisite software on Linux
  - Preparing your Linux operating system
  - Preparing the primary node on Linux

Perform the following steps to install and configure your Web server:

1. Install and configure the Web server; refer to the Web server documentation for information.
2. If using IBM Lotus® Domino®, edit the `NOTES.INI` file on the Web server. Set the `HTTPEnableConnectorHeaders` parameter to 1.
3. If using IBM HTTP Server or Apache Server, edit the `httpd.conf` file on the Web server. Set the `AllowEncodedSlashes` directive to `On`; the directive should be added at the root level as a global directive.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>See the appropriate HTTP Server documentation</td>
<td>IBM HTTP Server</td>
</tr>
<tr>
<td>See the appropriate Apache Server documentation</td>
<td>AllowEncodedSlashes directives</td>
</tr>
</tbody>
</table>

4. Stop the Web server.
5. Install and configure the Web server plug-in on the system where the Web server is located using the Plug-ins installation wizard provided with WebSphere Application Server. Refer to Selecting a Web server topology diagram and roadmap for information. **Note:** The WebSphere Application Server Web server instructions are for the default profile. If not using the default profile, refer to Configuring a Web server for a non-default profile.

   **If using WebDAV:** After successfully installing the Web server plug-in, locate and open your `plugin-cfg.xml` file and set `AcceptAllContent` to `true`.

   **Important:** Depending on how you use the Web server, you may need to adjust the `ServerIOTimeout` value, which defines how long the plug-in should wait for a response from the application. The recommended minimum value is 60 but you may need to adjust this value higher if you are retrieving data from a database. To update this value, locate and open your `plugin-cfg.xml` file and set `ServerIOTimeout` to a value that is appropriate for your business needs. For additional information, see Common questions about the Web server plug-in.

6. If you are using Sun Web Server Version 7 update 8, read Technote 1448262 and perform the steps to resolve the HTTP 408/409 error.
7. Start the Web server.
Preparation your Linux operating system

View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

- Prerequisites
  - Preparing prerequisite and corequisite software on Linux

Perform the following steps to prepare your Linux machine:

1. Perform the steps to prepare your operating system for the IBM WebSphere Application Server installation that comes with WebSphere Portal; see one of the following links for information.

   - WebSphere Application Server Version 6.1: Preparing Linux systems for installation
   - WebSphere Application Server Version 7.0: Preparing Linux systems for installation

2. Set the file descriptor limit to 10240; for example, `ulimit -n 10240`.

3. **Web Content Management only:** Use the `ulimit -f` command to set the maximum size of files that can be created to be at least the size of the largest file you would need to upload to the content server. The command `ulimit -f unlimited` removes any limit on file size.

4. Install and configure X server on Linux (for example X-Windows or GNOME) to use the graphical user interface the installation program provides. **Note:** X server is not required if installing with a response file or in console mode.

**Parent topic:** Setting up a cluster on Linux  
**Previous topic:** Preparing prerequisite and corequisite software on Linux  
**Next topic:** Preparing the primary node on Linux
Preparing the primary node on Linux

Before creating your high availability environment, you must install IBM® WebSphere® Portal on your primary node and then configure your database and your network deployment manager.

**Prerequisites**

- Preparing prerequisite and corequisite software on Linux
- Preparing your Linux operating system

Perform the following tasks to prepare your primary node:

1. **Installing WebSphere Portal on Linux on the primary node**
   IBM WebSphere Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

2. **Configure WebSphere Portal to use a remote database**
   For high availability, using a remote database is preferred. For improved performance databases can be distributed across multiple database servers. Databases should also be configured for failover. Configuring the database for failover is beyond the scope of this documentation. Refer to the database product documentation for the most authoritative guidance about setting up databases for fail over.

3. **Configuring the primary node to communicate with the deployment manager on Linux**
   After installing IBM WebSphere Application Server Network Deployment and installing IBM WebSphere Portal on the primary node, you must configure the primary node and the deployment manager to communicate with each other.

4. **Removing search collections on Linux**
   If you plan to use search in a cluster, you must configure a remote search server; see Configuring Portal Search in a cluster for information. If you created any search collections, you will need to recreate them on the remote search server. If your search collection has data, export the collection before deleting it and then import the data on the remote server.

**Parent topic:** Setting up a cluster on Linux

**Previous topic:** Preparing your Linux operating system

**Next topic:** Choosing the type of cluster to create on Linux
Installing WebSphere Portal on Linux on the primary node

IBM® WebSphere® Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

Review the Supported hardware and software requirements for your operating system and ensure that all hardware and software matches the minimum product level before installing WebSphere Portal. Review Installation methods, options, and sources before installing to determine your installation source location and installation method.

Perform the following steps to install WebSphere Portal and create a custom profile for the node that will be added to the cluster. If you already installed WebSphere Portal, you can skip to the last step to create the custom profile.

1. Type `ping yourserver.yourcompany.com` on a command line to verify that your fully qualified host name is properly configured.
2. Type `ping localhost` on a command line to verify that your network is properly configured.
3. If you are installing with an existing WebSphere Application Server instance, ensure it is installed at the supported level and has the following features installed:
   - **Restriction:** Installing WebSphere Portal Version 6.1.0.1 or higher on an existing WebSphere Application Server Version 7.0 instance where WebSphere Portal Version 7.0 was installed is unsupported.
   - **Important:** Besides ensuring that the existing WebSphere Application Server instance is at the supported level, you will also need to ensure that Apache Derby is installed at the supported level before installing WebSphere Portal. See WebSphere Portal detailed system requirements for supported levels.
   - Application Server
   - Administration
     - Scripted Administration
     - Administrative Console
   - Ant and Deployment Tools
     - Deploy Tool
     - Ant Utilities
4. Optional: Perform the following steps if you want to install WebSphere Portal using a non-root user:
   - **Restriction:** Although it is possible to install WebSphere Application Server as a non-root user, there are limitations to this option that can affect WebSphere Portal functionality, which is why you should install WebSphere Application Server as a root user.
   - A. Install the current supported version of WebSphere Application Server as a root user.
   - B. Open a command prompt and perform the following steps to create a non-root user and to change ownership:
     1. Use the appropriate system commands to create a new group, to create a new user, and to add the new user to the new group.
     2. Run the following tasks to change the rights of the non-root user:
        ```bash
        chmod -R g+rwx /opt/IBM
        chgrp -R group_name /opt/IBM
        chmod -R g+wr /tmp
        ```
C. Launch the installation program per the next step. During the installation, select the existing WebSphere Application Server using the non-root user and set the installation location to a unique location under the path where non-root permissions were granted.

5. Choose one of the following installation commands based on the installation method you want to use to install WebSphere Portal; see Installation Methods for more information:

**Advance installation parameters:** To customize your installation, you can add various parameters to your installation commands; for example, you can specify your port information. See Advanced installation parameters for information.

**Restriction:** When defining your installation path, the ONLY supported characters are ASCII letters [A-Z and a-z], numbers [0-9], slashes [/ or \. depending on your operating system], and the dash [-].

<table>
<thead>
<tr>
<th>Table 1. Installation task options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Installation type</strong></td>
</tr>
<tr>
<td><strong>Graphical user interface</strong></td>
</tr>
<tr>
<td><strong>Console mode</strong></td>
</tr>
<tr>
<td><strong>Silent install</strong></td>
</tr>
</tbody>
</table>

**Note:** If the installation program does not detect a WebSphere Application Server instance that you know exists, exit the installation program and pass the WebSphere Application Server instance location using the command line; for example, ./install.sh -W was.undetectedWas="/my/WAS/location".

**Note:** If the GUI or console mode installation program fails to detect ports for either WebSphere Application Server or WebSphere Portal, a warning message displays and the installer offers another chance to enter the values. If the silent installation fails to detect ports for either WebSphere Application Server or WebSphere Portal, the installer will exit.

6. Verify your installation was successful; access WebSphere Portal using the http://yourserver:yourport/wps/portal format. Run the ./ConfigEngine.sh list-server-ports -DWasPassword=password task from the wp_profile_root/ConfigEngine directory to generate the wp_profile_root/ConfigEngine/log/wp_PortMatrix.txt file that lists WebSphere Portal ports for your installation.

7. Optional: After the WebSphere Portal installation completes successfully, run the ./ConfigEngine.sh configure-express -DPortalAdminPwd=password -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, if you want to enable the sample IBM Lotus Web Content Management content, such as internet and intranet sample sites.

**Restriction:** Run the configure-express task before configuring your database, user registry, context root, security, etc. If you ran any tasks other than the install task, do not run this task.

**Note:** See http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_ovw.html for tutorials on how to use the sample content.

The sample content includes:

- Creates a group called contentAuthors; members of this group are given privileges to create content in the sample
Internet and intranet sites. Navigate to the Administration area and then click Access > Users and Groups.
- Creates two new Web Content Management Libraries: "Internet Web Content 6.1.0" and "Intranet Web Content 6.1.0".
- Navigate to the Administration area and then click Portal Content > Web Content Libraries.
- Adds a portlet filter and applies the filter to various portlets in the sample Internet and intranet sites. You can see the definition of the filter in the WebSphere Application Server Administration console and examining the custom resources under the Environment area.
- Creates two new theme policies: InternetStyle and IntranetStyle. These styles are applied to sample Internet and intranet sites. Navigate to the Theme Customizer and then select the style.
- Creates several portlet clones of the Web Content Management rendering portlet. These portlet clones are used on sample Internet and intranet sites.
- Creates two virtual portals with context roots of wps/portal/intranet and wps/portal/internet. These are the sample Internet and intranet sites. Go to http://yourserver:yourport/wps/portal/internet and http://yourserver:yourport/wps/portal/intranet to access them.
- Creates several sample credential slots, including "Default slot for E-mail", "Default slot for Feeds", "Default slot for Miscellaneous", "Default slot for Web Clipping", and "Default slot for Web Content Management". Navigate to the Administration area and then click Access > Credential Vault > Manage System Vault Slots.
- Modifies the portlet palette to include some new portlets. Click the Portlets link to see the portlet palette.

8. Optional: If you ran the configure-express task, the owner of the items in the Web content libraries containing the Internet and Intranet Site Template content will be listed as uid=xyzadmin,o=defaultWIMFileBasedRealm. To update the owner information for these items to correspond to your portal administrator ID, complete the following steps:
   A. Edit the wp_profile_root/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties file.
   B. Add the following line to the file:

   uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN

   Replace portal_admin_DN with the distinguished name of the portal administrator.
   C. Save your changes and close the file.
   D. Run the ./ConfigEngine.sh action-express-memberfixer -DPortalAdminPwd=password -DWasPassword=password -DModifyPortsServer=servername -DStartPort=starting port number task, located in the wp_profile_root/ConfigEngine directory.

9. Optional: Perform the following steps if you need to change the ports for WebSphere Application Server or WebSphere Portal:
   Note: The starting port parameter is required for a successful completion of the modify-ports-by-startport task.
   Once you specify a start port, this port becomes the base for assigning port values. The code increments this value as each port is assigned, which means that the WebSphere Portal ports will be assigned incrementally starting with the port defined with the modify-ports-by-startport task.
   A. Stop the server1 and WebSphere_Portal servers.
   B. Run one of the following commands for each server you need to change:

<table>
<thead>
<tr>
<th>Method</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting port number</td>
<td>./[ConfigEngine.sh modify-ports-by-startport -DWasPassword=password -DModifyPortsServer=servername -DStartPort=starting port number]</td>
</tr>
</tbody>
</table>
C. Restart the server1 and WebSphere_Portal servers.

**Note:** Sample port files are available on the Setup disc.

```
./ConfigEngine.sh modify-ports-by-portsfile -DWasPassword=password -DModifyPortsServer=servername -DPortsFile=full path to ports file
```

The following is an example of the information within a port file although the port values will be different based on your environment:

```
BOOTSTRAP_ADDRESS=10031
SOAP_CONNECTOR_ADDRESS=10033
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=10032
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=10025
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=10036
WC_adminhost=10027
WC_defaulthost=33344
DCS_UNICAST_ADDRESS=10029
WC_adminhost_secure=10039
WC_defaulthost_secure=10035
SIB_ENDPOINT_ADDRESS=10026
SIB_ENDPOINT_SECURE_ADDRESS=10037
SIB_MQ_ENDPOINT_ADDRESS=10030
SIB_MQ_ENDPOINT_SECURE_ADDRESS=10028
ORB_LISTENER_ADDRESS=10034
```

**Parent topic:** Preparing the primary node on Linux

**Next topic:** Configure WebSphere Portal to use a remote database

**Related concepts**

- Installation methods
- IBM Support Assistant Lite for WebSphere Portal

**Related reference**

- Advanced installation parameters
Configure WebSphere Portal to use a remote database

For high availability, using a remote database is preferred. For improved performance databases can be distributed across multiple database servers. Databases should also be configured for failover. Configuring the database for failover is beyond the scope of this documentation. Refer to the database product documentation for the most authoritative guidance about setting up databases for fail over.

Password considerations when transferring data manually

For security reasons, you should not store passwords in the wkplc.properties and wkplc_comp.properties. Edit each of the properties files prior to running a configuration task, inserting the passwords needed for that task. After the task has run, delete all passwords from each file. For more information, see Deleting passwords from configuration scripts. Alternatively, you can specify the password on the command line using the following syntax:

- Windows: ConfigEngine.bat task_name -Dpassword_property_key=password_value
- UNIX: ./ConfigEngine.sh task_name -Dpassword_property_key=password_value
- i5/OS: ConfigEngine.sh task_name -Dpassword_property_key=password_value

As with other properties, each password property must have the -D prefix and be set equal to (=) a value. If you have multiple properties in a single command, use a space character between each -Dproperty=value setting.

- Prerequisites
  - Installing WebSphere Portal on Linux on the primary node
  - Technotes for database connectivity issues

- Configuring WebSphere Portal to use DB2
  View information on installing and setting up DB2 to work with WebSphere Portal.

- Configuring WebSphere Portal to use DB2 for z/OS
  View information on installing and setting up DB2 for z/OS to work with WebSphere Portal.

- Configuring WebSphere Portal to use Oracle
  View information on installing and setting up Oracle to work with WebSphere Portal.

- Configuring WebSphere Portal to use Oracle RAC
  View information on installing and setting up Oracle RAC to work with WebSphere Portal.

- Configuring WebSphere Portal to use SQL Server
  View information on installing and setting up SQL Server to work with WebSphere Portal.

- Verifying databases
  After you configure IBM WebSphere Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.

Parent topic: Preparing the primary node on Linux
Previous topic: Installing WebSphere Portal on Linux on the primary node
Next topic: Configuring the primary node to communicate with the deployment manager on Linux

Related Information
Troubleshooting WebSphere Portal Version 6.1 databases
Configuring WebSphere Portal to use DB2

View information on installing and setting up DB2 to work with WebSphere Portal.

1. Installing DB2
   View information on installing DB2 for use with WebSphere Portal.

2. Create users
   View the steps to set up users for DB2 to work with WebSphere Portal.

3. Creating remote databases
   A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

4. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

5. Modifying database properties
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

6. Optional: Configuring JCR collation support
   View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

7. Configure WebSphere Portal to use DB2
   View the steps to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

8. Configuring DB2 for large file handling in Web Content Management
   If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the fullyMaterializeLobData property in the WebSphere Application Server administrative console.

9. Optional: Changing driver types
   WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Parent topic: Configure WebSphere Portal to use a remote database
Installing DB2

View information on installing DB2 for use with WebSphere Portal.

Before you begin:

- Make sure DB2 is set up with updated kernel parameters according to the recommendations in the DB2 Quick Beginnings guide at DB2 Technical Support.
- **(For Type 2 drivers only):** DB2 Connect must be installed on the same system as WebSphere Portal, and have the same name as the server profile name.
- When you install DB2 using the DB2 installation program, it automatically creates a DB2 administrative user with the correct operating system rights.

1. If you are using the JDBC driver in type 2 mode, configure your DB2 Connect client with the following commands. If you are using a remote database, complete this step separately from the WebSphere Portal installation.
   - `db2 update dbm cfg using tp_mon_name WAS`
   - `db2 update dbm cfg using spm_name $hostname`, where $hostname is the host name of WebSphere Portal.
   
   Because the default for spm_name is the hostname itself, specifying the hostname parameter is optional. If your hostname is more than eight characters, use empty double quotes (" "). For example, `db2 update dbm cfg using spm_name " "`.

2. Before installing DB2, log in with a user ID that has administrative authority. The system user ID and password must match the database user ID and password for the WebSphere Portal databases. If you do not use the default DB2 database user ID, or if you must access a remote database, create the system user ID before installing DB2.

To set up the environment for the database access:
A. The initialization script for this user (for example, user-home/.profile) must contain a call to the `db2profile` script in the db-home/sqllib directory.
B. After you create the system user ID and password for the DB2 installation, add the user ID to the DB2 administration group (such as `db2adm`) for that system.

   To ensure that the user is set up correctly, login with this user ID and start the DB2 command line processor by executing the command: `db2`. If the command line processor displays, the user ID is set up correctly.

3. To install DB2, follow the instructions that are provided with the DB2 documentation to install DB2 and the required fix pack.

**Parent topic:** Configuring WebSphere Portal to use DB2

**Next topic:** Create users

**Related Information**
- DB2 Technical Support
- DB2 Information Center
Create users

View the steps to set up users for DB2 to work with WebSphere Portal.

Before you begin: You should have completed Installing DB2.

- **Prerequisites**
  - Installing DB2

For all databases, use one user with administrative rights on the operating system and the DB2 installation. This user can be the database administrative user that is created automatically by the DB2 installation program. This user is the database configuration user that will be used for configuration tasks: creating database tables and performing database transfer. If you choose to have WebSphere Portal also create the databases, the database user should be given SYSADM rights. You only need to manually create an administrator ID when you do not want to use an existing DB2 administrator ID.

A common user name is `db2inst1`, but you can assign any user name as long as it has administrative access and follows the limitations listed here. Do not change the user name after creating it.

The limitations on user names are:
- User names can contain one to eight characters.
- Group and instance names can contain one to eight characters.
- Names cannot be any of the following: `users`, `admins`, `guests`, `public`, `local`
- Names cannot begin with: `IBM`, `SQL`, `SYS`
- Names cannot include accented characters.
- Create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Following are the basic steps to create new users and grant privileges. For more information, refer to the DB2 product documentation.

1. From the DB2 Control Center, expand the object tree until you find the DB Users folder.
2. Right-click the DB Users folder and select Create/Add. The Create DB User notebook opens.
3. In the User name field, type a user ID.
4. Assign an authority for the user.
5. Click OK.

**Parent topic:** Configuring WebSphere Portal to use DB2

**Previous topic:** Installing DB2

**Next topic:** Creating remote databases
Creating remote databases

A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

Before creating these databases, note the following information:

- Use IBM® DB2 Universal Database™ Enterprise Server Edition JDBC driver, JCC, type 4.
- If the DB2 JDBC type 4 driver is used, only the instructions on the remote server are required. You do not need to install the DB2 client software, and you do not need to complete the steps related to the DB2 client.
- The client software, DB2 Connect, must be correctly configured to connect to the remote DB2 server instance, for example, db2inst1.
- These instructions assume that a remote DB2 server and DB2 Connect are already installed and running. For information on installing DB2 Connect, see the DB2 product documentation.

**Prerequisites**

- Installing DB2
- Create users

To create a database, you must be a DB2 System Administrator with sufficient database privileges (SYSADM or at a minimum SYSCTRL).

1. Log in to the DB2 server system as a user with sufficient database privileges.
2. Initialize a DB2 command environment by opening a command prompt and typing `su - db2inst1`, where `db2inst1` is the database user. The prompt changes to your operating system shell prompt, for example: `$`. In this mode, you must type `db2` at the beginning of each command and use double quotation marks (``) around the entire command. The following example shows how to use the double quotes; it is not an actual command that you run:

   ```
   db2 "CREATE REGULAR TABLESPACE SMS PAGESIZE 4 K MANAGED BY SYSTEM USING ('sms') BUFFERPOOL SMSPOOL"
   ```

   **Note:** If you are using the Command Line Processor (CLP), refer to your DB2 documentation for details. The command prompt is `db2=>`. In this mode, commands can be entered without the `db2` prefix or the double quotation marks. However, the following steps assume you are not using the CLP and are entering commands from your operating system shell prompt, for example: `$`.

3. Run the following commands on the DB2 server system to configure the DB2 database instance:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>db2set DB2_RR_TO_RS=YES</code></td>
</tr>
<tr>
<td></td>
<td><code>db2set DB2_EVALUNCOMMITTED=YES</code></td>
</tr>
<tr>
<td></td>
<td><code>db2set DB2_INLIST_TO_NLJN=YES</code></td>
</tr>
<tr>
<td></td>
<td><code>db2 &quot;UPDATE DBM CFG USING query_heap_sz 32768&quot;</code></td>
</tr>
<tr>
<td></td>
<td><code>db2 &quot;UPDATE DBM CFG USING maxagents 500&quot;</code></td>
</tr>
<tr>
<td></td>
<td><code>db2 &quot;UPDATE DBM CFG USING sheapthsres 0&quot;</code></td>
</tr>
</tbody>
</table>

4. **(Important)** Failure to complete this step will result in an unsuccessful database transfer. Run the following commands on the DB2 server system to create the necessary databases:

   **Notes:**
   - Replace `dbname` with the actual name of the database. Run the commands and each time replace `dbname` with the actual values for release, community, customization, Java Content Repository, Feedback, and Likeminds. You will need to run the commands once for each database for a total of six times.
   - **Remember:** DB2 database names cannot
exceed eight characters. Therefore, consider using these database names: release, commun, custom, jcrdb, fdbkdb, and lmdb.

```
db2 "CREATE DB dbname using codeset UTF-8 territory us PAGESIZE 8192"
db2 "UPDATE DB CFG FOR dbname USING applheapsz 4096"
db2 "UPDATE DB CFG FOR dbname USING app_ctl_heap_sz 1024"
db2 "UPDATE DB CFG FOR dbname USING stmtheap 32768"
db2 "UPDATE DB CFG FOR dbname USING dbheap 2400"
db2 "UPDATE DB CFG FOR dbname USING locklist 1000"
db2 "UPDATE DB CFG FOR dbname USING logfilsiz 4000"
db2 "UPDATE DB CFG FOR dbname USING logprimary 12"
db2 "UPDATE DB CFG FOR dbname USING logsecond 20"
db2 "UPDATE DB CFG FOR dbname USING logbufsz 32"
db2 "UPDATE DB CFG FOR dbname USING avg_appls 5"
db2 "UPDATE DB CFG FOR dbname using AUTO_MAINT off"

Complete the following:

A. On the DB2 server system, run the following commands. This step is only required for the IBM Java Content Repository database (jcrdb).

  - `jcrdb` is the name of the database used to store user data and objects
  - `jcr` is the jcr user for `jcrdb`
  - `dbpassword` is the password for the jcr user for the `jcrdb`

```
db2 "CONNECT TO jcrdb USER jcr USING dbpassword"
db2 "CREATE BUFFERPOOL ICMLSFRQBP4 SIZE 1000 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL ICMLSVOLATILEBP4 SIZE 16000 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL ICMLSMAINBP32 SIZE 16000 PAGESIZE 32 K"
db2 "CREATE BUFFERPOOL ICMLSFREQBP4 SIZE 1000 PAGESIZE 4 K"
db2 "CREATE REGULAR TABLESPACE ICMLFQ32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLFQ32') BUFFERPOOL ICMLSMAINBP32"
db2 "CREATE REGULAR TABLESPACE ICMLNF32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLNF32') BUFFERPOOL ICMLSMAINBP32"
db2 "CREATE REGULAR TABLESPACE ICMVFPQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('ICMVFPQ04') BUFFERPOOL ICMLSVOLATILEBP4"
db2 "CREATE REGULAR TABLESPACE ICMSFQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('ICMSFQ04') BUFFERPOOL ICMLSFREQBP4"
db2 "CREATE REGULAR TABLESPACE CMBINV04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('CMBINV04') BUFFERPOOL CMBMAIN4"
db2 "CREATE SYSTEM TEMPORARY TABLESPACE ICMLSYSSTSPACE32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('icmlsysstspace32') BUFFERPOOL ICMLSMAINBP32"
db2 "CREATE SYSTEM TEMPORARY TABLESPACE ICMLSYSSTSPACE4 PAGESIZE 4 K MANAGED BY SYSTEM USING ('icmlsysstspace4') BUFFERPOOL ICMLSVOLATILEBP4"

```
db2 "DISCONNECT jcrdb"
db2 "TERMINATE"
```

B. For JDBC Type 2 connections only: On DB2 Connect, use a text editor to open the /etc/services file. If it does not specify the DB2 connection service port, add the following text to specify the port for the remote DB2 instance:

```
db2 -inst1port1/tcp # DB2 connection service port
```

where `db2inst1` is the name of the DB2 instance ID on the system, and `port1` with the actual port number that is
assigned to the DB2 connection service in your DB2 server installation. The connection service port on the DB2 Client system, WebSphere Portal server, must match the connection service port on the DB2 server. The ports should match by number but not necessarily by name.

C. Set up the correct service name by entering the following command on the DB2 server system:  
   `db2 "UPDATE DBM CFG USING svcename svce_name"` where `svce_name` is the connection service port name that is specified above.

D. For JDBC Type 2 connections only: On DB2 Connect, set DB2COMM to TCP/IP by using the `db2set` command:
   `db2set DB2COMM=tcpip`.

E. For JDBC Type 2 connections only: Catalog the TCP/IP node with the IP address of the remote database server on DB2 Connect:
   `db2 "catalog tcpip node remote_db_node_alias remote database_server_node server connection_service_port"` where:
   - `remote_db_node_alias` is the alias name of the database server that you are defining for the WebSphere Application Server node name. The alias name can contain one to eight characters.
   - `database_server_node` is the fully qualified host name of your database server system.
   - `connection_service_port` is the name of the DB2 connection service port that is configured in the `/etc/services` file on the database server system.

F. For JDBC Type 2 connections only, catalog the WebSphere Portal databases on DB2 Connect, where:
   - `remote_db_name_domain` is the cataloged name of the databases on the server system for each domain.
   - `domain_alias_name` is the database alias names that you are defining.
   - `remote_db_node_alias` is the name that was used previously when you cataloged the TCP/IP node in the previous step.

   The alias for each database must be different from the actual database name and can only contain up to eight characters:
   `db2 "catalog db remote_db_name_domain as domain_alias_name at node remote_db_node_alias"`
   `db2 "catalog db remote_db_name_release as release_alias_name at node remote_db_node_alias"`
   `db2 "catalog db remote_db_name_community as comm_alias_name at node remote_db_node_alias"`
   `db2 "catalog db remote_db_name_customization as cust_alias_name at node remote_db_node_alias"`
   `db2 "catalog db remote_db_name_fdbkdb as fdbkdb_alias_name at node remote_db_node_alias"`
   `db2 "catalog db remote_db_name_lmdb as lmdb_alias_name at node remote_db_node_alias"`
   `db2 "catalog db remote_db_name_jcrdb as jcrdb_alias_name at node remote_db_node_alias"`

G. For JDBC Type 2 connections only: Log out of DB2 Connect by entering `db2 "terminate"`.

H. For JDBC Type 2 connections only, on DB2 Connect, test your remote connection by issuing the following command in the DB2 command window:  
   `db2 "connect to alias_name user username using password"`, where `alias_name` is the alias name that you defined above, `username` is the database user, and `password` is the password assigned to the database user.

**Parent topic:** Configuring WebSphere Portal to use DB2  
**Previous topic:** Create users  
**Next topic:** Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties` file.
- The page size of table spaces used by WebSphere Portal must be 8192 bytes.
- For details on creating table spaces refer to the documentation for the database.

**Prerequisites**
- Installing DB2
- Create users
- Creating remote databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties` that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`
   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword IN and a space. For example: `community.COMP_INST.tablespace=IN COMM8KSPACE`
   Repeat this step for each domain that you are transferring.
4. Save and close `dbdomain.space_mapping.properties`.
5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - **Windows:** `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
   - **UNIX:** `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

**Parent topic:** Configuring WebSphere Portal to use DB2  
**Previous topic:** Creating remote databases  
**Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

Tip: For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Working with properties files:
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:
  - release.DbName = release
  - jcr.DbName = jcrdb
  - feedback.DbName = fdbkdb
  - likeminds.DbName = lmdb
  - community.DbName = commdb
  - customization.DbName = custdb

  - If you are using a remote database, enter the values for the remote server.
  - Use a forward slash (/) instead of a backslash (\).
  - There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
  - The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.

  - Depending on which database domain has to be configured, replace dbdomain with:
    - release
    - customization
    - community
    - jcr
    - feedback
    - likeminds

  - The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If `DbUser`, `DbUrl`, and `DbPassword` are not the same across domains, the value for `DataSourceName` must differ from the `DataSourceName` of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   - `wp_profile_root/ConfigEngine/properties/wkplc.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_comp.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties`

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In `wkplc_comp.properties`, most properties are repeated for each domain.

2. Use a text editor to open the properties file `wkplc_comp.properties` and modify the values to correspond to your environment.
   A. For `dbdomain.DbType`, type `db2`.
   B. For `dbdomain.DbName`, type the name of the WebSphere Portal domain database. **Notes:**
      - This value is also the database element in the `dbdomain.DbUrl` property.
      - This value is the TCP-IP alias for the database.
   C. For `dbdomain.DbSchema`, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   D. For `dbdomain.DataSourceName`, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For `dbdomain.DbUrl`, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:** The database element of this value should match the value of `DbName`.
   F. For `dbdomain.DbUser`, type the user ID for the database administrator.
   G. For `dbdomain.DbPassword`, type the password for the database administrator.
   H. Optional: For `dbdomain.DbRuntimeUser`, type the user ID of the database user that should be used by WebSphere Portal to connect to the database at runtime.
   I. If `dbdomain.DbRuntimeUser` is specified, you must set `dbdomain.DbRuntimePassword` to be the password of the runtime database user.
   J. For `dbdomain.DBA.DbUser`, type the database administrator user ID for privileged access operations during creation of the database.
K. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.

L. For `dbdomain.XDbName`, type the database loop back alias that needs to be set if you plan to use the `create-database` task. **Note:** Required only for local databases using a JDBC Type 2 driver.

M. For `dbdomain.DbNode`, type the value for the node database. Set this value if you want to call `create-database`. **Note:** Required only for local databases.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For `source.domainDbType`, type of the database you are currently configured to use. The value for `source.domainDbType` is Derby by default.

B. For `source.domainDbName`, type the name of the database domain you are currently using.

C. For `source.domainDbSchema`, type current schema identifier for objects within the database for this domain.

D. For `source.domainDataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For `source.domainDbUrl`, type the url currently used to access your database.

F. For `source.domainDbUser`, type the name of the user accessing this database.

G. For `source.domainDbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

   A. For `db2.DbDriver`, type the name of the JDBC driver class.

   B. For `db2.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For `db2.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.
Configuring JCR collation support

View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties

1. Stop the WebSphere Portal server:
   ```bash
   ./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password
   ```
2. If using a remote DB2 server, copy the following files from the WebSphere Portal server to a temporary directory on the remote DB2 server:
   ```bash
   PortalServer_root/jcr/prereq.jcr/config/collation.jar
   PortalServer_root/jcr/prereq.jcr/config/registerCollationUDFTemplate.sql
   ```
3. Set up collation on the database where the JCR domain is located.
   A. Change to the directory `db2_instance_owner_home/sqllib/function`.
   B. Execute the command:
      ```bash
      Remote DB2:
db2_instance_owner_home/sqllib/java/jdk/bin/jar -xvf temporary location/collation.jar icm/CollationUDF.class
      ```
      ```bash
      Local DB2:
db2_instance_owner_home/sqllib/java/jdk/bin/jar -xvf PortalServer_root/jcr/prereq.jcr/config/collation.jar icm/CollationUDF.class
      ```
   C. If using a local DB2, change to the directory `wp_profile_root/PortalServer/jcr/config`. If using a remote DB2, change to the temporary directory where you copied the files in a previous step.
   D. Open the file `registerCollationUDFTemplate.sql` and change all `SCHEMA` references to the JCR schema; for example, `JCR`. The value set for `SCHEMA` should match the value set for the `jcr.DbSchema` property. You specify `jcr.DbSchema` in the configuration file `wkplc_comp.properties` when you modify database properties.
   E. Connect to the JCR database by running `db2 connect to <jcrdb> user <userid> using <password>`. You must connect to the JCR database using the userid and password of the WebSphere Portal JCR data source.
   F. Execute the following command in the previous substep. If the command completes successfully, the UDF is registered correctly as follows:
      ```bash
      values schema.sortkeyj('abc','en'),
      ```
      where `schema` is the schema used in the previous substep.
   G. Disconnect from the JCR database.
   H. Restart the DB2 instance.
4. Verify that the UDF is registered properly.
   A. Log in as the `db2instanceID`. Open a DB2 terminal window, and type `db2`. From the command line, type the command `connect to JCRDB user userid using password`. You must connect to the JCR database using the userid and password of the WebSphere Portal JCR data source.
   B. Execute the following command in the previous substep. If the command completes successfully, the UDF is registered correctly as follows:
      ```bash
      values schema.sortkeyj('abc','en'),
      ```
      where `schema` is the schema used in the previous substep.
   C. Edit the `icm.properties` file, located in `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm` directory. Add the following section to the end of the file:
      ```bash
      # Enable/Disable collation support for all DB2 platforms
      ```

1622
# Disabled by default
jcr.query.collation.db2.enabled = true

# Database specific collation mappings
# These mappings apply map a Java locale name into a collation name
# supported by the underlying database.
# Example mappings for DB2 platform

# English
jcr.query.collation.en = en

# Swedish
jcr.query.collation.sv = sv

jcr.query.collation.zh = zh
jcr.query.collation.de = de
jcr.query.collation.da = da
jcr.query.collation.hu = hu
jcr.query.collation.jp = jp

6. Enter the following command to start the WebSphere Portal server:../startServer.sh WebSphere_Portal
Configure WebSphere Portal to use DB2

View the steps to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

**Prerequisites**

- Installing DB2
- Create users
- Creating remote databases
- Modifying database properties

**Tips:**

- To run these tasks as a non-root user, you must first run the task chown -R non-root_userWebSphereDir.
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

**Steps for transferring data to another supported database**

1. If you are running a type 2 connection, before transferring data edit the db2cli.ini file. Failure to follow these steps will cause the database transfer to hang at the task action-process-constraints.
   A. Locate the file /home/db2inst1/sqllib/cfg/db2cli.ini.
   B. Add the following to the end of the file. Leave an empty line after ReturnAliases=0.[COMMON]

   ```
   [COMMON]
   DYNAMIC=1
   ReturnAliases=0
   ```

2. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.

3. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2.

   ```
   /home/db2inst1/sqllib/db2profile
   ```

   where db2inst1 represents your database instance**Note:** You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.

4. Enter the following commands to validate configuration properties...

   ```
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ```

5. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.
6. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td><code>./stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td><code>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

7. Transfer the database:

   **Important:** Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory `wp_profile_root/ConfigEngine`.
   B. Enter the following command: `./ConfigEngine.sh database-transfer -DWasPassword=password`

      **Note:**
      - To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command: `./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password`
      - If you have been storing data in IBM® Derby for a long time, database transfer could fail with OutOfMemory exceptions. If database transfer fails, add the following property to the command in this step: `./ConfigEngine.sh database-transfer -DDbtJavaMaxMemory=1536M -DWasPassword=password`

   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

   If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

8. After transferring the database tables, perform a `reorg` check to improve performance. Perform this step for each database alias in the property file.

   A. Connect to a database with the following command: `db2 connect to database_alias user db2admin_userid using password`

      **Note:** Additional options might be required if additional security has been installed. Refer to DB2 Universal Database commands by example for links to the command reference.

   B. After it is connected, run the following commands from the DB2 prompt: `db2 reorgchk update statistics on table all > xyz.out`

   C. Look in the `reorg` column for entries marked with a * (star or asterisk) in the file `xyz.out`. For each line with a *, note the `tablename` and run the following command for each `tablename`: `db2 reorg table tablename db2 terminate db2rbind database_name -l db2rbind.out -u db2_admin -p password`

   D. The output file `db2rbind.out` is only created when there is an error for the `db2rbind` command.

9. Change to the directory `wp_profile_root/bin`.

10. Enter the following command to start the WebSphere Portal server: `./startServer.sh WebSphere_Portal`
Configuring DB2 for large file handling in Web Content Management

If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the fullyMaterializeLobData property in the WebSphere Application Server administrative console.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties
  - Configure WebSphere Portal to use DB2

**Note:** You only need to perform these steps if you are using Web Content Management.

1. Log into the WebSphere Application Server administrative console.
2. Click **Resources > JDBC > Data sources**.
3. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
4. Select the name of the data source that is defined in wkplc_comp.properties for the JCR database domain. The default data source is **wpdbDS**.
5. Click **Custom properties**.
6. Ensure that the fullyMaterializeLobData property is set to false.

**Parent topic:** Configuring WebSphere Portal to use DB2  
**Previous topic:** Configure WebSphere Portal to use DB2  
**Next topic:** Changing driver types
Changing driver types

WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Before you begin, ensure that the following conditions are met:

- The WebSphere Portal database has been successfully transferred to DB2 using the database-transfer configuration task.
- The files wkplc_comp.properties and wkplc_dbtype.properties have been modified to set the correct values for the DB2 drivers that you are switching to:

  In the file wkplc_comp.properties set each <Domain>.DbUrl property using the following formats:
  
  # db2 (type 2):        { jdbc:db2:wpsdb }
  # db2 (type 4):        { jdbc:db2://<YourDatabaseServer>:50000/wpsdb:returnAlias=0; }

  In the file wkplc_dbtype.properties set the db2.DbLibrary property using the following format:

  # For DB2 Type 2
driver use <SQLLIB>/java/db2java.zip
  # For DB2 Type 4 driver use <SQLLIB>/java/db2jcc.jar:<SQLLIB>/java/db2jcc_license_cu.jar

  In the file wkplc_dbtype.properties set the db2.DbDriver property using the following format:

  # For DB2 Type 2
driver use com.ibm.db2.jcc.DB2Driver
  # For DB2 Type 4 driver use com.ibm.db2.jdbc.app.DB2Driver

**Note:** If WebSphere Portal is installed on the same machine as the DB2 server and you switch from a JDBC Type 4 connection to a JDBC Type 2 connection, verify that you have created the alias names for the DB2 databases as described in Creating remote databases and that the alias names are specified for the databases in the file wkplc_comp.properties. When switching from a JDBC Type 2 connection to a JDBC Type 4 connection, remove the database alias names and refer to the databases directly. This is required because of a limitation in the DB2 Universal JDBC driver.

**Prerequisites**

- Installing DB2
- Create users
- Creating remote databases
- Modifying database properties
- Configure WebSphere Portal to use DB2
- Configuring DB2 for large file handling in Web Content Management

1. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.
2. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2.

   ```bash
   /home/db2inst1/sqllib/db2profile
   
   where db2inst1 represents your database instance
   **Note:** You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.

3. Enter the following commands to validate configuration properties.

   ```bash
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ```

Operating systems: AIX, HP-UX, i5/OS, Linux, Solaris, Windows
4. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

```
Option                     Description
WebSphere Application Server  ./stopServer.sh server1 -username admin_userid -password admin_password
WebSphere Portal             ./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password
```

6. Change to the directory `wp_profile_root/ConfigEngine`.

7. To change from one supported driver to the other, run the following task to connect the database, including only the domains that require the switch:

```
```

8. Change to the directory `wp_profile_root/bin`.

9. Enter the following command to start the WebSphere Portal server:

```
./startServer.sh WebSphere_Portal
```

**Parent topic:** Configuring WebSphere Portal to use DB2  
**Previous topic:** Configuring DB2 for large file handling in Web Content Management
Configuring WebSphere Portal to use DB2 for z/OS

View information on installing and setting up DB2 for z/OS to work with WebSphere Portal.

1. Installing DB2 for z/OS
   View information on how to install DB2 for z/OS for use with IBM® WebSphere® Portal.

2. Creating users
   Complete this procedure to set up users for IBM DB2 Universal Database™ for z/OS® to work with IBM WebSphere Portal.

3. Creating remote databases
   A remote database resides on a different machine than WebSphere Portal. You must manually create the required databases before configuring WebSphere Portal to work with DB2 for z/OS.

4. Optional: Creating the Java Content Repository database
   View information on setting up your Java Content Repository database to work with WebSphere Portal.

5. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

6. Configuring WebSphere Portal to use DB2 for z/OS
   View information on manually transferring data to the DB2 for z/OS you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository to DB2 for z/OS. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configure WebSphere Portal to use a remote database
Installing DB2 for z/OS

View information on how to install DB2 for z/OS for use with IBM® WebSphere® Portal.

The following prerequisites must exist separately from the WebSphere Portal installation:

- If you are using the DB2 Legacy Type 2 JDBC driver, configure your DB2 Connect client with the following commands:
  - `db2 update dbm cfg using tp_mon_name WAS`
  - `db2 update dbm cfg using spm_name hostname`, where `hostname` is the host name for the machine where the DB2 Connect client is installed (same as portal machine).

- If you are using the DB2 Legacy Type 2 JDBC driver, enable extended shared memory usage with the following commands:
  - `export EXTSHM=ON`
  - `db2set DB2ENVLIST=EXTSHM`
  - `db2start`

  To make the change permanent, you must add the environment variable by adding the following to the `profile.env` file:

  ```
  DB2ENVLIST='EXTSHM'
  ```

  in `/home/db2inst/sqllib/userprofile`

  add: `export EXTSHM=ON`

  **Note:** The shell must be reopened before restarting DB2 for z/OS.

- Both type 2 and type 4 drivers are supported. If you are using the DB2 Legacy Type 2 JDBC driver, the DB2 Connect client must be installed on the same machine as WebSphere Portal to connect to the remote database.

- The DB2 subsystem must be on a supported z/OS platform.

- Update the BP8K0 catalog bufferpool to 35,000 buffers before performing database transfer. You should change this value based on your environment. The SYSIBM.SYSDATABASE table resides in this bufferpool and is used extensively by DB2 for z/OS during database transfer.

To use DB2 for z/OS as the database software for WebSphere Portal, you must have DB2 for z/OS installed on your z/OS system. Refer to the DB2 for z/OS product documentation for instructions. Refer to the Planning for DB2 for z/OS topic for additional considerations for configuring DB2 for z/OS as the WebSphere Portal database.

**Parent topic:** Configuring WebSphere Portal to use DB2 for z/OS

**Next topic:** Creating users
Creating users

Complete this procedure to set up users for IBM® DB2 Universal Database™ for z/OS® to work with IBM WebSphere® Portal.

Before you begin: You should have completed Installing DB2 for z/OS.

- **Prerequisites**
  - Installing DB2 for z/OS

- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

- If you are configuring multiple WebSphere Portal instances to use a single DB2 for z/OS subsystem, you must create a unique database user for each WebSphere Portal instance. By default all database table names include the name of the database user used to access the data. Therefore, to prevent table name conflicts, you must create a unique database user for each WebSphere Portal instance on the shared DB2 for z/OS subsystem.

- Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Use the following steps to grant access permissions to the database users. Repeat these steps for each WebSphere Portal instance you are setting up.

1. Create all database user IDs in the security product you are using on z/OS. For jcrschema, the database schema name for IBM Java Content Repository data, create a group and connect it to the database user ID for Java Content Repository data, jcr. The following sample shows the RACF definition of such a user ID and group, where jcr is the database user ID for Java Content Repository data, yourDefaultUserGroup is your default RACF group for database user IDs, jcrschema is the database schema name for Java Content Repository data, and yourDefaultGroup is your default RACF group for groups. If you have some other security product such as Top Secret or ACF2 instead of RACF, translate the sample RACF definition into the appropriate syntax before running the job.

   ```
   ADDUSER jcr DFLTGRP(yourDefaultUserGroup) NAME('WAS DB2 ACCESS USER')
   PW USER(jcr) NOINTERVAL
   ALU jcr PASSWORD(********) NOEXPIRED
   ADDGROUP jcrschema SUPGROUP(yourDefaultGroup)
   CONNECT jcr GROUP(jcrschema)
   ```

2. Run the following SQL statements using a tool like SPUFI while logged on to the DB2 subsystem to grant appropriate rights on the newly created databases. If you are configuring multiple WebSphere Portal instances to use a single DB2 for z/OS subsystem, be sure to use the database user associated with the WebSphere Portal instance you are setting up. **Note:** Make sure your vmmdb user follows these same standards.

   WebSphere Portal instance you are setting up. (C) create/alter tablespaces
   (C) create/alter tables
   (C) create/alter indice;
   (C+R) read/write data

   (C) – at configuration time
   (R) – at runtime
GRANT USE OF ALL BUFFERPOOLS TO releaseusr;
GRANT USE OF ALL BUFFERPOOLS TO communityusr;
GRANT USE OF ALL BUFFERPOOLS TO customizationusr;

GRANT DBADM ON DATABASE jcrdbnameonzos TO jcr;
GRANT USE OF ALL BUFFERPOOLS TO jcr;
GRANT DBADM ON DATABASE fdbdbnameonzos TO feedback;
GRANT USE OF ALL BUFFERPOOLS TO feedback;
GRANT DBADM ON DATABASE lmddbnameonzos TO lmdbusr;
GRANT USE OF ALL BUFFERPOOLS TO lmdbusr;
GRANT SELECT ON SYIBM.SYSCOLUMNS TO releaseusr;
GRANT SELECT ON SYIBM.SYSCOLUMNS TO communityusr;
GRANT SELECT ON SYIBM.SYSCOLUMNS TO customizationusr;

GRANT SELECT ON SYIBM.SYSTABLES TO releaseusr;
GRANT SELECT ON SYIBM.SYSTABLES TO communityusr;
GRANT SELECT ON SYIBM.SYSTABLES TO customizationusr;

GRANT SELECT ON SYIBM.SYSCOLUMNS TO jcr;
GRANT SELECT ON SYIBM.SYSTABLES TO jcr;
GRANT SELECT ON SYIBM.SYSCOLUMNS TO fdbkbdbusr;
GRANT SELECT ON SYIBM.SYSTABLES TO fdbkbdbusr;
GRANT SELECT ON SYIBM.SYSCOLUMNS TO lmdbusr;
GRANT SELECT ON SYIBM.SYSTABLES TO lmdbusr;
GRANT SELECT ON SYIBM.SYSFOREIGNKEYS TO releaseusr;
GRANT SELECT ON SYIBM.SYSFOREIGNKEYS TO communityusr;
GRANT SELECT ON SYIBM.SYSFOREIGNKEYS TO customizationusr;

GRANT SELECT ON SYIBM.SYSFOREIGNKEYS TO jcr;
GRANT SELECT ON SYIBM.SYSFOREIGNKEYS TO fdbkbdbusr;
GRANT SELECT ON SYIBM.SYSFOREIGNKEYS TO lmdbusr;
GRANT SELECT ON SYIBM.SYSRELS TO releaseusr;
GRANT SELECT ON SYIBM.SYSRELS TO communityusr;
GRANT SELECT ON SYIBM.SYSRELS TO customizationusr;

GRANT SELECT ON SYIBM.SYSRELS TO jcr;
GRANT SELECT ON SYIBM.SYSRELS TO fdbkbdbusr;
GRANT SELECT ON SYIBM.SYSRELS TO lmdbusr;
GRANT USE OF STOGROUP jcrstogroup TO jcr;
GRANT CREATEIN, DROPIN ON SCHEMA jcrschema TO jcr;
GRANT SELECT ON SYIBM.SYSTABLESPACE TO jcr;
GRANT SELECT ON SYIBM.SYSVIEWS TO jcr;
GRANT SELECT ON SYIBM.SYSDUMMY1 TO jcr;
GRANT SELECT ON SYIBM.SYSTRIGGERS TO jcr;
GRANT SELECT ON SYIBM.SYSINDEXPART TO jcr;
GRANT SELECT ON SYIBM.SYSINDEXES TO jcr;
GRANT SELECT ON SYIBM.SYSSYNONYMS TO jcr;

where:
3. releasenameonzos, communitynameonzos, customizationnameonzos, and releaseusr, communityusr, customizationusr represent the databases and database users, respectively, of the WebSphere Portal instance you are setting up. (These users must be created on the host system.)

- jcrdbnameonzos and jcr are the database and database user, respectively, for Content Repository data.
- fbdbdbnameonzos and feedback are the database and database user, respectively, for Feedback data.
- lmdbnameonzos and lmdbusr are the database and database user, respectively, for Likeminds data.

- jcrschema is the database schema name for Content Repository data.

3. Grant the necessary access rights to all users who might require them. Depending on the architecture that you choose, these users might include Java Content Repository and Feedback users.

Parent topic: Configuring WebSphere Portal to use DB2 for z/OS
Previous topic: Installing DB2 for z/OS
Next topic: Creating remote databases
Creating remote databases

A remote database resides on a different machine than WebSphere Portal. You must manually create the required databases before configuring WebSphere Portal to work with DB2 for z/OS.

Before you begin, complete the following prerequisites:

- These instructions assume that you will use a local IBM® DB2 Universal Database™ Enterprise Server Edition Connect to connect to a remote IBM DB2 Universal Database for z/OS® server, and that IBM WebSphere® Application Server, WebSphere Portal, and DB2 Connect will be installed on the same machine.

- To run SQL statements, you can use a tool like SPUFI. The specified statements include CREATE STOGROUP, CREATE DATABASE and CREATE TABLESPACE. For CREATE STOGROUP you have to replace the icmvolumes and icmvcat variables with volume serial numbers and a catalog name for your environment. For CREATE TABLESPACE you can specify a specific BUFFEROOL. Refer to the DB2 for z/OS SQL Reference for a description of these options.

- Ensure that a TEMP database is created for your subsystem. You can use the following statements to create a TEMP database:

```
CREATE DATABASE db_name AS TEMP;
CREATE TABLESPACE ts_name IN db_name;
```

- Log on to the DB2 subsystem on the host server. DB2 system administrator rights are needed to create the databases.

- Replace variables as follows:

  - releasenameonzos, communitynameonzos, and customizationnameonzos are the WebSphere Portal databases for WebSphere Portal and Member Manager data.
  - fdbkdbnameonzos and fdbkdbts are the database and table space, respectively, for Feedback data.
  - lmdbnameonzos and lmdbts are the database and table space, respectively, for LikeMinds data.

- Because large objects are stored in columns that can become very large, logging changes to these columns requires a huge amount of log space. For this reason, large object (LOB) logging is disabled by default for tablespaces that contain such data. With LOB logging disabled, you can recover full backups, but not incremental backups that can be used for point in time recovery. To recover point in time backups, you must enable LOB logging. For detailed instructions, see Technote 1306637, Managing LOB logging in DB2 for z/OS.

- Prerequisites

  - Installing DB2 for z/OS
  - Creating users

Use the following steps to set up WebSphere Portal with a remote instance of DB2 for z/OS. Refer to Planning for DB2 for z/OS for a list of databases and table space names. If you are configuring multiple WebSphere Portal instances to use a single DB2 subsystem, be sure to use the database and table space names associated with the WebSphere Portal instance you are setting up.

1. CREATE DATABASE releasenameonzos CCSID UNICODE;
2. CREATE DATABASE communitynameonzos CCSID UNICODE;
3. CREATE DATABASE customizationnameonzos CCSID UNICODE;
4. Execute the steps in the topic Creating the Java Content Repository database.
5. CREATE DATABASE fdbkdbnameonzos CCSID UNICODE;
6. CREATE TABLESPACE fdbkdbts IN fdbkdbnameonzos USING STOGROUP SYSDEFLT PRIQTY 5000 SECQTY 500 LOCKSIZE ROW;
7. CREATE DATABASE lmdbnameonzos CCSID UNICODE;
8. CREATE TABLESPACE `lmdbts` IN `lmdbnameonzos` USING STOGROUP SYSDEFLT PRIQTY 5000 SECQTY 500;

Related Information

- Managing LOB logging in DB2 for z/OS
Creating the Java Content Repository database

View information on setting up your Java Content Repository database to work with WebSphere Portal.

First, you need to create multiple databases for scalability.

The IBM Java Content Repository database grows in size as the IBM® WebSphere® Portal server is used. To support this growth, which includes the dynamic creation of many new tables within the database, the WebSphere Portal server allows you to create multiple databases within a single IBM DB2 Universal Database™ for z/OS® subsystem. A minimum of two databases is recommended, but more can be created if necessary. A single database setup is not recommended; it will quickly become constrained. See the Performance guides on the product documentation Web site for more information.

Each database that is created must begin with a common prefix; there is no convention required for the remainder of the name. For example, to create \textit{xx} number of databases, you may choose to use the following commands:

\begin{verbatim}
CREATE DATABASE JCRDB01
CREATE DATABASE JCRDB02
... 
CREATE DATABASE JCRDBxx
\end{verbatim}

In this case, \textit{JCR} is the common prefix. You also have to select a maximum number of user-defined tables (UDT) that each database will be allowed to hold; the recommended value is 400.

\textbf{- Prerequisites}

- Installing DB2 for z/OS
- Creating users
- Creating remote databases

A sample script of the commands that you can use to create the Java Content Repository database in your DB2 for z/OS installation is shown below. The sample demonstrates the creation of a single database. To follow the recommendation of creating at least two databases, duplicate the commands for each additional database as indicated below. Find a template of these commands in the file \texttt{PortalServer\_root/installer/wp.config/config/templates/db/db2\_zos/jcr\_sample.sql}.

\textbf{Notes:}

- It is not necessary to duplicate every tablespace definition in every database.
- In order to run the commands shown in the sample, you must know the database name and the IDs of the MVS volumes where the Java Content Repository database tables will be stored.
- Edit the template file for your installation environment before running the commands shown in the sample:
  - Replace \texttt{jcrdbnameX} with the name of the database. Remember that each database name must begin with a common prefix.
  - Replace \texttt{stogroup} with the name of your storage group.
  - Replace \texttt{icmvolumes} with the IDs of the MVS volumes where the Java Content Repository database tables will be stored. These values are assigned by the database administrator.
  - Replace \texttt{icmvcat} with the name of the virtual catalog.
  - Replace \texttt{jcr} with the name of database user ID.
  - Replace \texttt{4kbp} with the name of your 4K bufferpool.
  - Replace \texttt{32kbp} with the name of your 32K bufferpool.
  - Replace \texttt{jcrschema} with the schema name of your Java Content Repository domain.
CREATE STOGROUP stogroup VOLUMES (icmvolumes) VCAT icmvcat
GRANT USE OF STOGROUP stogroup to jcr

CREATE DATABASE jcrdbnameX STOGROUP stogroup BUFFERPOOL 4kbp INDEXBP 4kbp CCSID UNICODE

Note: The following two tablespace commands must be created in every database:
CREATE TABLESPACE ICMLFQ32 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICMTSQ04 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 5000 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

Note: The following tablespaces are required in the first database only:
CREATE TABLESPACE ICVFQ04 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 5000 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICMSFQ04 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 2000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSTADO IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRIDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 2000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRSCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRISLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRGCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRIGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSTDPV IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSDOAC IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RESMLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COLNLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USERLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USEGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE SYSCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICNLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USERSCLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCCLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSTNLS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USERCLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USEGLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCLCLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSTOLSL IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USERCLLSLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USEGLLSLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCLCLLSLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSTPLS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USERCLPLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USEGLPLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCCLPLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE SYSTEMPLS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSTPLLSL IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USERCLPLLSLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USEGLPLLSLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCLCLPLLSLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CACLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CPERLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NLSLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE MAXKLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NLSKLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITTDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITVDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITVILSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COMDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COMALSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COMFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COVDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COVALSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITALLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE IT11LSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE IV11LSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE LI11LSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITTRLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RMATLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CHEOLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE MIMTLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITEELSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE
1639


CREATE TABLESPACE ICUT301 IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE ICMACLS IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE ICMACLC IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE ICMACTS IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE TS00208 IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE TS00207 IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE TS00206 IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE TS00205 IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE TS00204 IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE TS00203 IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE TS00210 IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE TS00202 IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE TS00209 IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE TS00201 IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE TS00200 IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE TIELLSTS IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE REPLLSTS IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE REPRLSTS IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE RI11LSTS IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE XDOFLSTS IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE XDOOLSTS IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE IDELLSTS IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE TEICLSTS IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE SYAELSTS IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE HSUDDLS IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE VERSCH IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE DBCURR IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE JCLCURR IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE TASKS IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE JCLLOGS IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE JCLNGS IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE JCLCURR IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE JCLCURR IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE JCLCURR IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE JCLCURR IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE JCLCURR IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE JCLCURR IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE JCLCURR IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE JCLCURR IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL

CREATE TABLESPACE JCLCURR IN 
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TSPEJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TSTAJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RIDSJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE WSNJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE LOB TABLESPACE ICMLOBT1 IN jcrdbnameX LOG NO
CREATE LOB TABLESPACE ICMLOBT2 IN jcrdbnameX LOG NO
CREATE LOB TABLESPACE ICMLOBT3 IN jcrdbnameX LOG NO
CREATE ALIAS jcrschema.ICMFICTITIOUS FOR SYSIBM.SYSDUMMY1
CREATE TABLESPACE DLKRJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE LKRLJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

Parent topic: Configuring WebSphere Portal to use DB2 for z/OS
Previous topic: Creating remote databases
Next topic: Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the \wp_profile_root\PortalServer\config\tablespaces\dbdomain.space_mapping.properties file.
- The page size of table spaces used by WebSphere Portal must be 8192 bytes.
- For details on creating table spaces refer to the documentation for the database.

**Prerequisites**
- Installing DB2 for z/OS
- Creating users
- Creating remote databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file \wp_profile_root\PortalServer\config\tablespaces\dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.index_name.indexspace
   For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword IN and a space. For example: community.COMP_INST.tablespace=IN COMM8KSPACE
   Repeat this step for each domain that you are transferring.
4. Save and close dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
   - UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
- `i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
Configuring WebSphere Portal to use DB2 for z/OS

View information on manually transferring data to the DB2 for z/OS you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository to DB2 for z/OS. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

- Prerequisites
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases

Steps for transferring data to another supported database

1. If you are running a type 2 connection, before transferring data edit the `db2cli.ini` file. Failure to follow these steps will cause the database transfer to hang at the task `action-process-constraints`.
   A. Locate the file `/home/db2inst1/sqllib/cfg/db2cli.ini`.
   B. Add the following to the end of the file. Leave an empty line after `ReturnAliases=0`.

   ```
   [COMMON]
   DYNAMIC=1
   ReturnAliases=0
   ```

2. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.

3. Enter the following commands to validate configuration properties...

   ```
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```

4. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td><code>./stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td><code>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

6. Transfer the database: **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory `wp_profile_root/ConfigEngine`.
   B. Enter the following command: `./ConfigEngine.sh database-transfer -DWasPassword=password`

   **Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter...
the following command: 

```
./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
```

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files and then repeat this step.

7. From the command line processor, reset the check pending status on the WebSphere Portal table spaces listed here. CHECK DATA is a DB2 batch utility that is invoked through JCL or through the DB2 utility panels. Type the following utility commands to check pending status:

```
check data tablespace releasenameonzos.TS320A
check data tablespace releasenameonzos.TS280A
check data tablespace releasenameonzos.TS300A
check data tablespace releasenameonzos.TS2110A
check data tablespace releasenameonzos.TS830A
check data tablespace communitynameonzos.TS8000B
check data tablespace communitynameonzos.TS8011B
check data tablespace communitynameonzos.TS280B
check data tablespace customizationnameonzos.TS2110C
check data tablespace jcrdbnameonzos.ICMSFQ04
check data tablespace jcrdbnameonzos.TS2110D
```

where releasenameonzos, communitynameonzos, and customizationnameonzos are the names of your WebSphere Portal databases, and jcrdbnameonzos is the name of your JCR database. Refer to the Utility Guide and Reference for DB2 for z/OS for additional details.

8. Run the RUNSTATS utility as shown in the following example:

```
LISTDEF JCRDBZOS INCLUDE TABLESPACE JCRDBZOS.* BASE
RUNSTATS TABLESPACE LIST JCRDBZOS INDEX(ALL) KEYCARD TABLE(ALL)
LISTDEF JCRDB01 INCLUDE TABLESPACE JCRDB01.* BASE
RUNSTATS TABLESPACE LIST JCRDB01 INDEX(ALL) KEYCARD TABLE(ALL)
...```

9. Start the WebSphere Application Server.

10. Verify that the WebSphere Portal application server is running by opening the following URL in a browser:

```
http://hostname.companyname.com:port_number/wps/portal
```

where hostname.companyname.com is the fully qualified host name of the machine where WebSphere Portal is running and port_number is the transport port that is created by WebSphere Application Server.

If you have performed database transfer in a clustered environment, you must explicitly copy the `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm/icm.properties` file from the primary node on which the database-transfer task was processed to all secondary nodes. This ensures that the secondary nodes have the new JCR database values specified in the icm.properties file.

1. Stop the portal server on the secondary node.
2. Copy the `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm/icm.properties` file from the primary node and replace the icm.properties file on the secondary node with the new file from the primary node.
3. Start the portal server on the secondary node.
Configuring WebSphere Portal to use Oracle

View information on installing and setting up Oracle to work with WebSphere Portal.

1. Installing Oracle
   View information on how to install Oracle for use with WebSphere Portal.

2. Creating users
   View the steps to set up users for Oracle to work with WebSphere Portal.

3. Creating databases
   View some important considerations before setting up Oracle databases to work with WebSphere Portal.

4. Modifying database properties
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

6. Creating JCR table spaces
   This topic provides manual instructions for creating JCR table spaces for Oracle.

7. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

8. Configure WebSphere Portal to use Oracle
   This section provides information on how to manually transfer data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configure WebSphere Portal to use a remote database
Installing Oracle

View information on how to install Oracle for use with WebSphere Portal.

You can use Oracle as the database software. You must install Oracle separately and before performing a database transfer.

Refer to Oracle product documentation for installation instructions.

Parent topic: Configuring WebSphere Portal to use Oracle

Next topic: Creating users
Creating users

View the steps to set up users for Oracle to work with WebSphere Portal.

Before you begin:

- You should have installed Oracle.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for
  WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

**Prerequisites**

- Installing Oracle

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work
with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example,
avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Start the Oracle SQL command line processor 'SQL*Plus' by entering `sqlplus /nolog` on the operating system
   command prompt

2. Log in to the Oracle database by executing command `connect user/password@database name`. For example:
   `system/manager@wpsdb` will log the administrative user system with a password of manager into the wpsdb database.

3. Create the WebSphere Portal user `dbdomain.DbUser`, where `dbdomain` is replaced by release, community, or
   customization.

   A. Log in to the database in which you want to create the new users.

   B. Create the users that you need in the appropriate databases by entering the following commands in the SQL*Plus
      tool, where `user_tablespace` is the default tablespace that is identified by the database administrator to store user
      objects and `temp_tablespace` is identified to store temporary objects. All names must be in lower case.

      ```
      SQL> create user releaseusr identified by password
      default tablespace user_tablespace
      temporary tablespace temp_tablespace;
      GRANT UNLIMITED TABLESPACE TO releaseusr;
      SQL> create user communityusr identified by password
      default tablespace user_tablespace temporary tablespace temp_tablespace;
      GRANT UNLIMITED TABLESPACE TO communityusr;
      SQL> create user customizationusr identified by password
      default tablespace user_tablespace temporary tablespace temp_tablespace;
      GRANT UNLIMITED TABLESPACE TO customizationusr;
      SQL> create user jcr identified by password
      default tablespace user_tablespace temporary tablespace temp_tablespace;
      GRANT UNLIMITED TABLESPACE TO jcr
      ```

4. Connect to the Feedback database:

   A. Enter the following command: `SQL> connect`
B. Enter user-name:  username/password@dbname where username is an existing administrative user in the database. For example, system/manager@fdbkdb will log the administrative user system with a password of manager into the fdbkdb database.

C. Create the Feedback user:SQL> create user feedback identified by password
default tablespace user_tablespace
temporary tablespace temp_tablespace;

GRANT UNLIMITED TABLESPACE TO feedback

D. Log out of the command line tool using the command SQL> exit.

5. Connect to the LikeMinds database:
   A. Enter the following command:SQL> connect
   B. Enter user-name:  username/password@dbname, where username is an existing administrative user in the database.
      For example, system/manager@lmdb will log the administrative user system with a password of manager into the lmdb database.

C. Create the LikeMinds user:SQL> create user lmdbusr identified by password
default tablespace user_tablespace
temporary tablespace temp_tablespace;

GRANT UNLIMITED TABLESPACE TO lmdbusr

D. Log out of the command line tool using the command SQL> exit.

6. Perform the following steps to verify that you can connect to the database with the user ID and password that you just created:
   A. Enter the SQL> connect command to connect to the content database.
   B. Enter user-name:username/password@dbname, where username is an existing administrative user in the database. For example, system/manager@jcrdb will log the administrative user system with a password of manager into the Oracle database.

Parent topic: Configuring WebSphere Portal to use Oracle
Previous topic: Installing Oracle
Next topic: Creating databases
Creating databases

View some important considerations before setting up Oracle databases to work with WebSphere Portal.

For information about creating databases, refer to the Oracle product documentation. For information on the recommended database architecture and the databases you will need to create, see the Planning for Oracle topic. Be sure that all databases to be used with WebSphere Portal are created as UNICODE character set databases.

If you are using remote Oracle 9 or 10 databases, you must also copy the ojdbc14.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM® WebSphere® Application Server Version 7.0, you must also copy the ojdbc6.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM WebSphere Application Server Version 6.1, you must also copy the ojdbc5.jar file from the remote Oracle server to the WebSphere Portal machine. The typical location is the oracle_home/sqldveloper/jdbc/lib directory. Record the copy location on your local machine for future reference.

When creating Oracle databases for use with WebSphere Portal, you should consider the following information:

- The Oracle databases must be created manually before configuring WebSphere Portal.
- All databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle server will be remote or local to the WebSphere Portal installation.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle database in order for WebSphere Portal to communicate with the Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes:

  - db_block_size = 8192 bytes
  - db_cache_size = 307,200 bytes
  - db_files = 1024 files
  - log_buffer = 65536 bytes
  - open_cursors = 1500 cursors
  - pga_aggregate_target = 204,800 bytes
  - pre_page_sga = true
  - processes = 300 processes
  - shared_pool_size = 204,800 bytes

  **Note:** If you are using IBM Java Content Repository, the open_cursors value may need to be increased based on the table count in the Java Content Repository schema.

- Raise the number of parallel servers as appropriate. For example, if you have more than 875 parallel servers, you should set the parallel_max_servers to 1200.
- The Oracle parameter CURSOR_SHARING allows similar SQL Statements to be shared when possible, which prevents parsing and establishing a new execution plan. The execution plan is used by Oracle to gather the data needed to satisfy a request. There are two options for CURSOR_SHARING, which are as follows:
  - **FORCE**
    - When you select this option, Oracle uses the same execution plan for all SQLs that are similar in value even if the values are different. When you use this option, the execution plan may not provide optimum performance. For example, similar SQLs with different values may behave differently when executed running the same plan.
  - **EXACT**
When you select this option, Oracle only shares the same execution plan for SQLs that are identical and use the same values. This option removes the risk of a SQL statement being executed when optimum performance conditions do not exist.

WebSphere Portal supports both options. Regardless of the option selected, portlet applications should not be affected. Contact your database administrator for further assistance on these options.

- **Prerequisites**
  - Installing Oracle
  - Creating users

Parent topic: Configuring WebSphere Portal to use Oracle
Previous topic: Creating users
Next topic: Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Installing Oracle
  - Creating users
  - Creating databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**

- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as **release.DbName**, **jcr.DbName**, **feedback.DbName**, and **likeminds.DbName**. For example:
  - `release(DbName)=release`
  - `jcr(DbName)=jcrdb`
  - `feedback(DbName)=fdbkdb`
  - `likeminds(DbName)=lmdb`
  - `community(DbName)=commdb`
  - `customization(DbName)=custdb`

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.

- Depending on which database domain has to be configured, replace `dbdomain` with:
  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`

- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.
   A. For dbdomain DbType, type oracle.
   B. For dbdomain DbName, type the name of the WebSphere Portal domain database. Note: This value is also the database element in the dbdomain DbUrl property.
   C. For dbdomain DbSchema, type the schema name of the database domain. Note: Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The dbdomain.DbName should be the same value used for the dbdomain.DbSchema
      Restriction: The value for dbdomain.DbSchema must equal the value for dbdomain.DbUser.
   D. For dbdomain DataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For dbdomain DbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. Note:
      - The database element of this value should match the value of DbName.
      - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following:
        jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.
F. For \textit{dbdomain}.DbUser, type the user ID for the database administrator. **Restriction:** The value for \textit{dbdomain}.DbUser must equal the value for \textit{dbdomain}.DbSchema.

G. For \textit{dbdomain}.DbPassword, type the password for the database administrator.

H. For \textit{dbdomain}.DBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.

I. For \textit{dbdomain}.DBA.DbPassword, type the database administrator password for privileged access operations during creation of the database.

J. For \textit{dbdomain}.DbHome, type the root location for the database. **Note:** This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

   A. For \textit{source.domain}.DbType, type of the database you are currently configured to use. The value for \textit{source.domain}.DbType is Derby by default.

   B. For \textit{source.domain}.DbName, type the name of the database domain you are currently using.

   C. For \textit{source.domain}.DbSchema, type current schema identifier for objects within the database for this domain.

   D. For \textit{source.domain}.DataSourceName, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

   E. For \textit{source.domain}.DbUrl, type the url currently used to access your database.

   F. For \textit{source.domain}.DbUser, type the name of the user accessing this database.

   G. For \textit{source.domain}.DbPassword, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file \textit{wkplc_dbtype.properties}.

   A. For oracle.DbDriver, type the name of the class that SqlProcessor uses to import SQL files.

   B. For oracle.DbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For oracle.JdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file \textit{wkplc.properties}.

   A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Configuring WebSphere Portal to use Oracle

**Previous topic:** Creating databases

**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

- **Prerequisites**
  - Installing Oracle
  - Creating users
  - Creating databases
  - Modifying database properties

1. On the database server, make sure that the folder `db_1` contains the subfolders `your_oracle_instance/data` and `your_oracle_instance/index`. If this folder hierarchy does not exist, create it manually before you run the `setup-database` task. The `setup-database` task requires these folders to create database users. If these folders do not exist, the `setup-database` task will fail.

2. Change to the directory `wp_profile_root/ConfigEngine`

3. To create the database users, type the following command:

   ```bash
   ./ConfigEngine.sh setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:

   ```sql
   '<WP_root>/base/wp.db.impl/config/templates/setupdb/sqlserver2005\<database domain>\createUsers.sql'.
   ```

**Parent topic:** Configuring WebSphere Portal to use Oracle

**Previous topic:** Modifying database properties

**Next topic:** Creating JCR table spaces

**Related tasks**

Manually creating users and granting privileges for Oracle
Creating JCR table spaces

This topic provides manual instructions for creating JCR table spaces for Oracle.

- **Prerequisites**
  - Installing Oracle
  - Creating users
  - Creating databases
  - Modifying database properties
  - Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.
2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:
   - `$jcrdb` is the name of the database you created to store user data.
   - `$dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.
   
   Ensure that the `.` is included in the variables when you substitute the values of your environment with these variables.

   **Important:** You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

   ```sql
   create tablespace ICMLFQ32 datafile '$dbpath./jcrdb./data/ICMLFQ32_01.dbf' size 300M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLNF32 datafile '$dbpath./jcrdb./data/ICMLNF32_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMVFQ04 datafile '$dbpath./jcrdb./data/ICMVFP04_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMSFQ04 datafile '$dbpath./jcrdb./data/ICMSFP04_01.dbf' size 150M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLSNDX datafile '$dbpath./jcrdb./index/ICMLSNDX_01.dbf' size 10M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   
   A. Set the `size`, `autoextend`, and `maxsize` values according to your environment. For example, you may want to change the `maxsize` to a set value rather than `UNLIMITED`.
   B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.
   C. Refer to the Oracle command reference for more information about using the `create tablespace` command.

Parent topic: Configuring WebSphere Portal to use Oracle

Previous topic: Setting up databases

Next topic: Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root`/PortalServer/config/tablesspaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

**- Prerequisites**

- Installing Oracle
- Creating users
- Creating databases
- Modifying database properties
- Setting up databases
- Creating JCR table spaces

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tablesspaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`

   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds

3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword TABLESPACE and a space. For example: `community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE`

   Repeat this step for each domain that you are transferring.

4. Save and close `dbdomain.space_mapping.properties`.
5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
- UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

**Parent topic:** Configuring WebSphere Portal to use Oracle  
**Previous topic:** Creating JCR table spaces  
**Next topic:** Configure WebSphere Portal to use Oracle
Configure WebSphere Portal to use Oracle

This section provides information on how to manually transfer data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- **Prerequisites**
  - Installing Oracle
  - Creating users
  - Creating databases
  - Modifying database properties
  - Setting up databases
  - Creating JCR table spaces

**Tips:**
- If you are transferring from Oracle, the `open_cursors` setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

**Steps for transferring data to another supported database**

1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.
2. Enter the following commands to validate configuration properties:

   `./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password`

   `./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password`

3. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.
4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td>./stopServer.sh server1 -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>
5. Transfer the database. **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory `wp_profile_root/ConfigEngine`.
   
   B. Enter the following command:
   
   ```bash
   ./ConfigEngine.sh database-transfer -DWasPassword=password
   ```
   
   **Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:
   ```bash
   ./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
   ```
   
   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example:
   ```sql
   SQL> execute dbms_stats.gather_schema_stats(ownname=> 'jcr', cascade=> TRUE);
   ```

7. Change to the directory `wp_profile_root/bin`.

8. Enter the following command to start the WebSphere Portal server:
   ```bash
   ./startServer.sh WebSphere_Portal
   ```

**Parent topic:** Configuring WebSphere Portal to use Oracle

**Previous topic:** Assigning custom table spaces
Configuring WebSphere Portal to use Oracle RAC

View information on installing and setting up Oracle RAC to work with WebSphere Portal.

1. Installing Oracle RAC
   View information on how to install Oracle RAC for use with WebSphere Portal.

2. Create users
   View the steps to set up users for Oracle RAC to work with WebSphere Portal.

3. Creating databases
   View some important considerations before setting up Oracle RAC databases to work with WebSphere Portal.

4. Modifying database properties
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

6. Creating JCR table spaces
   This topic provides manual instructions for creating JCR table spaces for Oracle.

7. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

8. Configuring WebSphere Portal to use Oracle RAC
   This section provides information on how to manually transfer data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configure WebSphere Portal to use a remote database
Installing Oracle RAC

View information on how to install Oracle RAC for use with WebSphere Portal.

Before you begin:
- You should have completed reviewing the Planning for Oracle RAC section.
- Ensure the Oracle CRS (Cluster Ready Service) and Oracle RAC databases have been installed and configured on the primary and secondary nodes.
- Run the following commands on both RAC nodes to start global services daemon_(GSD), oracle listeners, and agents.
  - `$ gsdctl start`
  - `$ lsnrctl start`
  - `$ agentctl start`

You can use Oracle RAC as the database software for WebSphere Portal. You must install Oracle RAC separately and before performing WebSphere Portal database transfer.

Refer to Oracle product documentation for installation instructions.

Parent topic: Configuring WebSphere Portal to use Oracle RAC
Next topic: Create users
Create users

View the steps to set up users for Oracle RAC to work with WebSphere Portal.

Before you begin:
- You should have completed installing Oracle RAC.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Create the users in the WebSphere Portal databases
   A. Log in to the database in which you want to create the new users.
   B. Create the users that you need in the appropriate databases by entering the following command in the SQL*Plus tool, where `user_tablespace` is the default tablespace that is identified by the database administrator to store user objects and `temp_tablespace` is identified to store temporary objects:

   ```
   SQL> create user username identified by password
   default tablespace user_tablespace
   temporary tablespace temp_tablespace;
   ```

   **Tip:** Balance the storage of user objects among tablespaces to prevent running out of space with overuse of `user_tablespace`. Also consider increasing the size of `user_tablespace` when handling a high volume of users.

2. Log in by entering the command `$ sqlplus` in SQL*Plus:

3. Enter the command `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example: `system/manager@wpsdb` will log the administrative user system with a password of manager into the wpsdb database.

4. Create the WebSphere Portal user `dbdomain.DbUser`, where `dbdomain` is replaced by release, community, customization, or JCR. All names must be in lower case. The following code uses the default user tablespace and the default temporary tablespace. Explicitly specify these tablespaces if you do not want to use the defaults:

   ```
   SQL> create user releasusr identified by password;
   SQL> create user communityusr identified by password;
   SQL> create user customizationusr identified by password;
   SQL> create user jcr identified by password;
   ```

5. Connect to the content database by entering the command `SQL> connect`.

6. Enter the following, where `username` is an existing administrative user in the database. `user-name:username/password@dbname` For example, `system/manager@jcrdb` will log the administrative user system with a password of manager into the Oracle RAC database.

7. Create the Oracle RAC users by entering the following: When creating the `jcr` user, grant all necessary privileges. If you do not grant privileges, you will receive the error `ICMADMIN lacks CREATE SESSION Privilege logon denied when you try to connect with the jcruser`.

   ```
   SQL> create user jcr identified by password default tablespace users temporary tablespace temp;
   ```
8. Connect to the Feedback database:
   A. Enter the following command: `SQL> connect` 
   B. Enter `user-name: username/password@dbname` where `username` is an existing administrative user in the database. For example, `system/manager@fdbkdb` will log the administrative user `system` with a password of `manager` into the `fdbkdb` database.

9. Create the Feedback user: `SQL> create user feedback identified by password default tablespace users temporary tablespace temp;`

10. Connect to the LikeMinds database: `SQL> connect` 
11. Enter `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example, `system/manager@lmdb` will log the administrative user `system` with a password of `manager` into the `lmdb` database.

12. Create the LikeMinds user: `SQL> create user lmdbusr identified by password default tablespace users temporary tablespace temp;`

13. Log out of the command line tool using the command `SQL> exit`.

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC  
**Previous topic:** Installing Oracle RAC  
**Next topic:** Creating databases
Creating databases

View some important considerations before setting up Oracle RAC databases to work with WebSphere Portal.

When creating Oracle RAC databases for use with WebSphere Portal, you should consider the following information:

- The Oracle RAC databases must be created manually before configuring WebSphere Portal.
- All databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle RAC server will be remote or local to the WebSphere Portal installation.
- If using an earlier version of Oracle RAC (9i or earlier), ensure that Oracle RAC JVM is also installed.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle RAC database in order for WebSphere Portal to communicate with the IBM Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes are:

  - db_block_size = 8192 bytes
  - db_cache_size = 314,572,800 bytes
  - db_files = 1024 files
  - log_buffer = 65536 bytes
  - open_cursors = 1500 cursors
  - pga_aggregate_target = 209,715,200 bytes
  - pre_page_sga = true
  - processes = 300 processes
  - shared_pool_size = 209,715,200 bytes

  **Note:** If you are using IBM Java Content Repository, the open_cursors value may need to be increased based on the table count in the IBM Java Content Repository schema.

- **Prerequisites**
  - Create users

  Refer to the following instructions to create tablespaces:

  1. In the database directory, create the data directory data and the index directory index.
  2. Create the tablespaces. **Important:** You must use the same tablespace names listed in the commands. The tablespace names cannot be customized or modified.

    A. Find and edit the SQL script jcr_ora_tablespaces.sql in the directory wp_profile_root
       ./ConfigEngine/work/db/oracle.

    B. In the define section, replace the following variables with the values from your environment:

       - jcrdb
         - Name of the database you created to store user data.

       - logfile
         - Location to store the log file.

       - dbpath
         - Directory where you created the database.

    C. Set the size, autoextend, and maxsize values according to your environment. For example, you may want to change the maxsize to a set value rather than UNLIMITED. Consult your Database Administrator for more info.
D. Execute the following SQL script:

Notes:
- The DBA or a user with sufficient credentials (CREATE tablespace) must execute the script.
- The script will prompt for the database username and password.

```
# sqlplus
SQL > @jcr_ora_tablespaces.sql
```

Parent topic: Configuring WebSphere Portal to use Oracle RAC
Previous topic: Create users
Next topic: Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Create users
  - Creating databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:
  - `release.DbName=release`
  - `jcr.DbName=jcrdb`
  - `feedback.DbName=fbdkdb`
  - `likeminds.DbName=lmdb`
  - `community.DbName=commdb`
  - `customization.DbName=custdb`
- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace `dbdomain` with:
  - `release`  
  - `customization` 
  - `community`  
  - `jcr` 
  - `feedback`  
  - `likeminds`
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
  - `dbdomain.DbType`
- `dbdomain.DbName`
- `dbdomain.DbUrl`
- `dbdomain.DbSchema`

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If `DbUser`, `DbUrl`, and `DbPassword` are not the same across domains, the value for `DataSourceName` must differ from the `DataSourceName` of the other domains. In other words, this value must be unique for the database domain.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

1. Locate the following files and create a backup copy of each before changing any values:
   - `wp_profile_root/ConfigEngine/properties/wkplc.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_comp.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties`

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In `wkplc_comp.properties`, most properties are repeated for each domain.

2. Use a text editor to open the properties file `wkplc_comp.properties` and modify the values to correspond to your environment.

   A. For `dbdomainDbType`, type `oracle`.
   
   B. For `dbdomainDbName`, type the name of the WebSphere Portal domain database. Note: This value is also the database element in the `dbdomainDbUrl` property.
   
   C. For `dbdomainDbSchema`, type the schema name of the database domain. Note: Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The `dbdomainDbName` should be the same value used for the `dbdomainDbSchema`

   **Restriction:** The value for `dbdomainDbSchema` must equal the value for `dbdomainDbUser`

   D. For `dbdomainDataSourceName`, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:

   - releaseDS
   - communityDS
   - customizationDS
   - jcrDS
   - lmdbDS
   - feedback

   E. For `dbdomainDbUrl`, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. Note:

   - The database element of this value should match the value of `DbName`.
   - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following:
     `jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME`. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.

   F. For `dbdomainDbUser`, type the user ID for the database administrator. **Restriction:** The value for `dbdomainDbUser` must equal the value for `dbdomainDbSchema`. 

   1671
G. For dbdomain.DbPassword, type the password for the database administrator.

H. For dbdomain.DBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.

I. For dbdomain.DBA.DbPassword, type the database administrator password for privileged access operations during creation of the database.

J. For dbdomain.DbHome, type the root location for the database. **Note:** This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For source.domain.DbType, type of the database you are currently configured to use. The value for source.domainDbType is Derby by default.

B. For source.domain.DbName, type the name of the database domain you are currently using.

C. For source.domain.DbSchema, type current schema identifier for objects within the database for this domain.

D. For source.domain.DataSourceName, type the name of the datasourse that is currently used in your IBM WebSphere Application Server configuration.

E. For source.domain.DbUrl, type the url currently used to access your database.

F. For source.domain.DbUser, type the name of the user accessing this database.

G. For source.domain.DbPassword, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file wkplc_ddtype.properties.

A. For oracle.DbDriver, type the name of the class that SqlProcessor uses to import SQL files.

B. For oracle.DbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

C. For oracle.JdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file wkplc.properties.

A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC  
**Previous topic:** Creating databases  
**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

Before you begin, ensure that the following prerequisites are met:
- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

**Prerequisites**
- Create users
- Creating databases
- Modifying database properties

1. On the database server, make sure that the folder `db_1` contains the subfolders `your_oracle_instance/data` and `your_oracle_instance/index`. If this folder hierarchy does not exist, create it manually before you run the `setup-database` task. The `setup-database` task requires these folders to create database users. If these folders do not exist, the `setup-database` task will fail.
2. Change to the directory `wp_profile_root/ConfigEngine`
3. To create the database users, type the following command:

   ```
   ./ConfigEngine.sh setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:

   '``WP_root\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql''.
Creating JCR table spaces

This topic provides manual instructions for creating JCR table spaces for Oracle.

- Prerequisites
  - Create users
  - Creating databases
  - Modifying database properties
  - Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.

2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:
   - `&jcrdb` is the name of the database you created to store user data.
   - `&dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.

   Ensure that the `.` is included in the variables when you substitute the values of your environment with these variables.

   **Important:** You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

   ```sql
   create tablespace ICMLFQ32 datafile '/dbpath./&jcrdb/data/&jcrdb._ICMLFQ32_01.dbf' size 300M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLNF32 datafile '/dbpath./&jcrdb/data/&jcrdb._ICMLNF32_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMVFQ04 datafile '/dbpath./&jcrdb/data/&jcrdb._ICMVFP04_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMSFQ04 datafile '/dbpath./&jcrdb/data/&jcrdb._ICMSFQ04_01.dbf' size 150M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLSNDX datafile '/dbpath./&jcrdb/index/&jcrdb._ICMLSNDX_01.dbf' size 10M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   ```

   A. Set the `size`, `autoextend`, and `maxsize` values according to your environment. For example, you may want to change the `maxsize` to a set value rather than `UNLIMITED`.

   B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.

   C. Refer to the Oracle command reference for more information about using the `create tablespaces` command.

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC  
**Previous topic:** Setting up databases  
**Next topic:** Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:

- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

**Prerequisites**
- Create users
- Creating databases
- Modifying database properties
- Setting up databases
- Creating JCR table spaces

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`
   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
     - release
     - community
     - customization
     - jcr
     - feedback
     - likeminds
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword `TABLESPACE` and a space. For example: `community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE`
   Repeat this step for each domain that you are transferring.
4. Save and close `dbdomain.space_mapping.properties`.
5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - Windows: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
- UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC  
**Previous topic:** Creating JCR table spaces  
**Next topic:** Configuring WebSphere Portal to use Oracle RAC
Configuring WebSphere Portal to use Oracle RAC

This section provides information on how to manually transfer data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- **Prerequisites**
  - Create users
  - Creating databases
  - Modifying database properties
  - Setting up databases
  - Creating JCR table spaces

**Tips:**

- If you are transferring from Oracle, the `open_cursors` setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

- The WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following: `jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME`. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.

- Manually specify an RAC datasource URL in the WebSphere Application Server console. The URL should look like the following:
  
  ```
  jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT=1521)) (ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICE_NAME))).
  ```

- When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

**Steps for transferring data to another supported database**

1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.
2. Enter the following commands to validate configuration properties:

```bash
./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
```

3. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.

4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td><code>./stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td><code>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

5. Transfer the database:

- **Note**: Do not execute the database-transfer task as a background process. This might cause the task to stall.

A. Change to the directory `wp_profile_root/ConfigEngine`.

B. Enter the following command:

```
./ConfigEngine.sh database-transfer -DWasPassword=password
```

**Note**: To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:

```
./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
```

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example:

```
SQL> execute dbms_stats.gather_schema_stats(ownname=> 'jcr', cascade=> TRUE);
```

7. Specify the JDBC URL to connect to the cluster:

A. Login to the WebSphere Application Server Administrator Console

B. Navigate to Resources > JDBC Providers

C. If there is a value in the Node field, remove it and click Apply.

D. For each Oracle JDBC provider, repeat the following steps:

1. Click the provider name.
2. Select Data Sources.
3. Click the name of the data source.
4. Under the Oracle data source properties, modify the URL attribute according to Oracle RAC configuration:

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT=1521))) (ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICENAME))
```

5. Save your changes

8. Change to the directory `wp_profile_root/bin`.

9. Enter the following command to start the WebSphere Portal server:

```
./startServer.sh WebSphere_Portal
```

**Parent topic**: Configuring WebSphere Portal to use Oracle RAC

**Previous topic**: Assigning custom table spaces
Configuring WebSphere Portal to use SQL Server

View information on installing and setting up SQL Server to work with WebSphere Portal.

1. Installing SQL Server
   View the steps to install SQL Server for use with WebSphere Portal.

2. Modifying database properties
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

3. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

4. Optional: Assigning custom filegroups
   The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

5. Configuring WebSphere Portal to use SQL Server 2005
   This section provides information on how to manually transfer data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configure WebSphere Portal to use a remote database
Installing SQL Server

View the steps to install SQL Server for use with WebSphere Portal.

Before you begin this task, complete the following prerequisites:

- You should have completed reviewing the Planning for SQL Server topic.
- You must install SQL Server separately from WebSphere Portal.
- WebSphere Portal requires either the DataDirect Connect for JDBC 3.7 JDBC driver or the Microsoft SQL Server JDBC Driver.
- You can obtain the DataDirect Driver from DataDirect Technologies. The JDBC driver consists of two parts: the JDBC driver libraries and JTA libraries. The JDBC driver libraries must be put on the WebSphere Portal system, while the JTA libraries are intended for SQL Server.
- You can obtain the Microsoft SQL Server JDBC Driver from Microsoft. See the Microsoft SQL Server product documentation for installation details.

The following section provides instructions for installing SQL Server for use with IBM® WebSphere® Application Server and WebSphere Portal. These steps are the same for both the DataDirect and Microsoft drivers unless noted.

1. Install SQL Server and all required patches.
2. Select the authentication mode for this installation:
   - Windows Authentication Mode
   - Mixed Mode (Windows Authentication and SQL Server Authentication)
     Either authentication mode allows the user who is specified as the database administrator in the wkplc_comp.properties file to log in to the SQL server. Mixed Mode authentication allows either a Windows user or a SQL server user, or both, to log in to the SQL Server.
3. In the SQL Server Setup panel, **Components to Install**, select the following components, which are required services for WebSphere Portal:
   - **SQL Server Database Services**
     - **Integration Services** The option **Integration Services**, creates the directory DTS/Binn, where you will need to copy files when you install Microsoft SQL Server JDBC drivers and enable XA connections.
4. Complete the installation using SQL Server documentation as a guide.
5. Enable TCP/IP connectivity in the **SQL Server Configuration Manager**.
6. Install the JDBC driver using one of these methods:
   - Installing DataDirect Connect for JDBC drivers on UNIX
   - Installing DataDirect Connect for JDBC drivers on Windows
   - Installing Microsoft SQL Server JDBC drivers and enabling XA connections
7. Network the connections for MSDTC in the Windows Component Services as described in Microsoft Help and Support.
8. Start SQL Server.

**Installing DataDirect Connect for JDBC drivers on UNIX**

1. Purchase and download DataDirect Connect for JDBC and save file 360connectjdbc.jar in a temporary work directory.
To create the required files, run the following command from the directory that contains 360connectjdbc.jar:

```
jar -xvf 360connectjdbc.jar
```

3. Run `./Installer.sh` in the same directory.

4. When installation is complete, change directory to `/installation/lib`, where `installation` is the directory in which you installed the DataDirect Connect for JDBC drivers.

5. Run the following command to change the permissions on the JAR files in the `installation/lib` directory:

```
chmod 777 *.jar
```

6. Change the ownership and group of these files by running the following commands in the `installation/lib` directory:

```
chgrp system_grp *.jar
chown root *.jar
```

Where `system_grp` is the system group as labeled by your operating system.

### Installing Microsoft SQL Server JDBC drivers and enabling XA connections

1. Download and install the Microsoft SQL Server JDBC driver.

2. Copy file `sqljdbc_xa.dll` from the `xa` subdirectory to the `C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqljdbc_xa.dll` directory of the SQL Server installation.

3. Start the database server.

4. Ensure that the Distributed Transaction Coordinator has been started. The status can be verified in the list of services in the Computer Management console.

5. Start the Microsoft SQL Server Management Studio and connect to the local database engine as the system administrator, `sa`.

6. Select **File > Open > File** and select `xa_install.sql` from the subdirectory of the downloaded and extracted JDBC driver.

7. Execute the script by selecting **Query > Execute**. **Note:** Any warnings that appear in the messages section of the application window that say that stored procedures cannot be found can be safely ignored.

8. For Microsoft SQL Server JDBC drivers: If you are running Windows XP SP2, Windows XP 64-Bit Edition, or Windows Server 2003, refer to the Registry Entries Are Required for XA Transaction Support document for information on a new security constraint and how to set SQL Server on Windows XP SP2, Windows XP 64-Bit Edition, or Windows Server 2003. Create a additional value in the Windows registry for WebSphere Portal by following these steps:

   A. Open the Windows Registry Editor (`regedit`) and navigate to the element `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSDTC\XADLL`

   B. From the menu bar, select **Edit > New > String Value** to create a new parameter named `sqljdbc_xa.dll` in that element.

   C. Change the value of the new parameter to the location of the `sqljdbc_xa.dll` file copied in Step 2 above, for **example**: `C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqljdbc_xa.dll`

Parent topic: Configuring WebSphere Portal to use SQL Server

Next topic: Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Installing SQL Server

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:
  - `release.DbName` = `release`
  - `jcr.DbName` = `jcrdb`
  - `feedback.DbName` = `fdbkdb`
  - `likeminds.DbName` = `lmdb`
  - `community.DbName` = `commdb`
  - `customization.DbName` = `custdb`

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace `dbdomain` with:
  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
  - `dbdomain.DbType`
- dbdomain.DbName  
- dbdomain.DbUrl  
- dbdomain.DbSchema  

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.
   A. For dbdomainDbType, type sqlserver2005.
   B. For dbdomain.DbName, type the name of the WebSphere Portal domain database. Note: This value is also the database element in the dbdomain.DbUrl property.
   C. For dbdomain.DbSchema, type the schema name of the database domain. Note: Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   D. For dbdomain.DataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For dbdomain.DbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. Note: The database element of this value should match the value of DbName.
   F. For dbdomain.DbUser, type the user ID for the database administrator.
   G. For dbdomain.DbPassword, type the password for the database administrator.
   H. For dbdomain.DBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.
   I. For dbdomain.DBA.DbPassword, type the database administrator password for privileged access operations during creation of the database.
   J. For dbdomain.DbHome, type the root location for the database. Note: This value is the location to store the database files locally.
   K. For dbdomain.AdminUrl, type the sqlserver URL without a database attached. This value is used to connect to the server for database administration operations.
   L. For dbdomain.DbHostName, type the hostname of the database.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. Important: The default values for the following parameters are
set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For source.domainDbType, type of the database you are currently configured to use. The value for source.domainDbType is Derby by default.

B. For source.domainDbName, type the name of the database domain you are currently using.

C. For source.domainDbSchema, type current schema identifier for objects within the database for this domain.

D. For source.domainDataSourceName, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For source.domainDbUrl, type the url currently used to access your database.

F. For source.domainDbUser, type the name of the user accessing this database.

G. For source.domainDbPassword, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file wkplc_dbtype.properties.

A. For sqlserver2005.DbDriver, type the name of the class that SqlProcessor uses to import SQL files.

B. For sqlserver2005.DbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

C. For sqlserver2005.JdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

D. For sqlserver2005.DbConnectionPoolDataSource, type the name of the implementation class of the connection pool data source.

6. Save and close the file.

7. Update the following property in the file wkplc.properties.

A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

Before you begin, ensure that the following prerequisites are met:
- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

- **Prerequisites**
  - Installing SQL Server
  - Modifying database properties

1. Change to the directory `wp_profile_root/ConfigEngine`
2. To create the databases, type the following command:
   ```bash
   ./ConfigEngine.sh create-database -DWasPassword=password
   ```
3. To create the database users, type the following command:
   ```bash
   ./ConfigEngine.sh setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:

   `<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql`.

**Parent topic:** Configuring WebSphere Portal to use SQL Server

**Previous topic:** Modifying database properties

**Next topic:** Assigning custom filegroups

**Related tasks**
Create users and databases for SQL Server 2005 on AIX and UNIX
Assigning custom filegroups

The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

Before you begin:

- The custom filegroups must exist prior to the execution of the database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties` file.
- For details on creating filegroups refer to the documentation for the database.

**Prerequisites**

- Installing SQL Server
- Modifying database properties
- Setting up databases

If custom filegroups are assigned, each must be assigned explicitly. The default filegroups can be used to contain database objects; however the name of the default file group must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom filegroups:

1. Determine the names of your custom filegroups.

2. Open the mapping file `wp_profile_root/PortalServer/config/tablespaces` that specifies the table space and index space for each property pairs for each database table:

   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.indexspace.indexspace`

   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds

3. Assign a filespace to each entry in the mapping file. The filegroup name must be prepended by the keyword `ON` and a space. For example: `community.COMP_INST.tablespace=ON COMM8KSPACE`

   Repeat this step for each domain that you are transferring.

4. Save and close `dbdomain.space_mapping.properties`.

5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,

   - Windows: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
   - UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
   - i5/OS: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
Configuring WebSphere Portal to use SQL Server 2005

This section provides information on how to manually transfer data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- Prerequisites
  - Installing SQL Server
  - Modifying database properties
  - Setting up databases

Tips:
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

Steps for transferring data to another supported database

1. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.
2. Enter the following commands to validate configuration properties:
   ```bash
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```
3. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.
4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
   | WebSphere Application Server| ```bash
   .stopServer.sh server1 -username admin_userid -password admin_password
   ``` |
   | WebSphere Portal            | ```bash
   .stopServer.sh WebSphere_Portal -username admin_userid -password admin_password
   ``` |
5. Transfer the database: **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.
   A. Change to the directory wp_profile_root/ConfigEngine.
B. Enter the following commands:

```
./ConfigEngine.sh database-transfer -DWasPassword=password
```

**Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:

```
./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
```

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. Change to the directory `wp_profile_root/bin`.

7. Enter the following command to start the WebSphere Portal server:

```
./startServer.sh WebSphere_Portal
```

8. Update the SQL Server 2005 statistics for Portal, and JCR databases by opening SQL Server Management Studio, selecting `New Query`, and running the following query:

```
use db_name
exec sp_updatestats @resample='resample'
```

**Parent topic:** Configuring WebSphere Portal to use SQL Server

**Previous topic:** Assigning custom filegroups
Verifying databases

After you configure IBM® WebSphere® Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.

- Prerequisites
  - ▶️ Technotes for database connectivity issues

You can verify the database connection using IBM WebSphere Application Server or by opening WebSphere Portal in a browser.

- To verify that the WebSphere Portal application server is running by using WebSphere Application Server, complete these steps:
  1. Open the WebSphere Application Server administrative console by entering the following address in a browser:
     http://hostname.example.com:10027/ibm/console
     where hostname.example.com is the fully qualified host name of the machine where WebSphere Portal is running and 10027 is the default transport port that is created by WebSphere Application Server.
  2. Log into the administrative console.
  3. Depending on your version of WebSphere Application Server, click the appropriate option:
     - For WebSphere Application Server Version 6.1: Click Resources & JDBC Providers.
     - For WebSphere Application Server Version 7.0: Click Resources > JDBC > JDBC Providers
  4. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
  5. Select the name of the data source that is defined in wkplc_comp.properties. The default data source is wpdbDS.
  6. Select the name of the JDBC provider that is specified in wkplc_dbtype.properties. The default JDBC provider is wpdbJDBC_dbtype, where dbtype is replaced by the value that matches your environment.
  7. Click Test Connection to verify the database connection. If configuration parameters have been changed, you might need to restart WebSphere Application Server for the test to complete.

- To verify that the WebSphere Portal application server is running by opening WebSphere Portal in a browser, enter the following URL in a supported browser:
  http://hostname.example.com:10040/wps/portal
  where hostname.example.com is the fully qualified host name of the machine where WebSphere Portal is running and 10040 is the default transport port that is created by WebSphere Application Server.

Parent topic: Configure WebSphere Portal to use a remote database
Configuring the primary node to communicate with the deployment manager on Linux

After installing IBM® WebSphere® Application Server Network Deployment and installing IBM WebSphere Portal on the primary node, you must configure the primary node and the deployment manager to communicate with each other.

- **Prerequisites**
  - Installing WebSphere Portal on Linux on the primary node
  - Configure WebSphere Portal to use a remote database

Perform the following steps to configure WebSphere Portal to communicate with the deployment manager:

1. Run the `.ConfigEngine.sh collect-files-for-dmgr -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory of the primary node, to create an archive or compressed file containing all the files which need to be copied to the Deployment Manager. **Note:** The archive or compressed file will be placed in the `wp_profile_root/filesForDmgr` directory and the file will be called `filesForDmgr.zip`.

2. Stop the deployment manager.

3. Expand the `filesForDmgr.zip` file into the installation root directory of the Deployment Manager; for example in the `opt/IBM/WebSphere/AppServer` directory. **Note:** If the Deployment Manager profile was not created in the default `AppServer/profiles/Dmgr01` directory, then the metadata_wkplc.xml file, located in the `AppServer/profiles/Dmgr01/config/.repository/metadata_wkplc.xml` directory in the zip file, must be placed into the correct Deployment Manager profile directory.

4. If the Deployment Manager profile is running on the same application server where WebSphere Portal is installed, remove the `com.ibm.ws.portletcontainer.deploytask_6.1.5.jar` file from the `AppServer\plugins` directory.

5. Start the deployment manager.

**Parent topic:** Preparing the primary node on Linux

**Previous topic:** Configure WebSphere Portal to use a remote database

**Next topic:** Removing search collections on Linux

**Related tasks**

Configuring Portal Search in a cluster on Linux
Removing search collections on Linux

If you plan to use search in a cluster, you must configure a remote search server; see Configuring Portal Search in a cluster for information. If you created any search collections, you will need to recreate them on the remote search server. If your search collection has data, export the collection before deleting it and then import the data on the remote server.

Prerequisites
- Installing WebSphere Portal on Linux on the primary node
- Configure WebSphere Portal to use a remote database
- Configuring the primary node to communicate with the deployment manager on Linux

Perform the following step to remove the search collection:

Perform the following steps to delete all existing search collections from the primary node:
1. Log on to WebSphere Portal.
2. Navigate to Administration > Search Administration > Manage Search and then click Search Collections.
3. Click the Delete Collection icon for each search collection and then click OK until they are all deleted.
4. Restart the WebSphere_Portal server and then navigate back to the Search Collections page to verify that all search collections have been deleted.

Parent topic: Preparing the primary node on Linux
Previous topic: Configuring the primary node to communicate with the deployment manager on Linux
Choosing the type of cluster to create on Linux

If you installed a IBM® WebSphere® Application Server Network Deployment, create a static cluster to handle failover requests. If you installed a IBM WebSphere Virtual Enterprise, create a dynamic cluster to balance member workloads.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on Linux
  - Preparing your Linux operating system
  - Preparing the primary node on Linux

Choose one of the following options to create a cluster:

- **Creating a static cluster on Linux**
  After installing IBM WebSphere Portal on the primary node, configuring a remote database, and preparing the primary node to communicate with the Deployment Manager, you can create your static cluster to handle failover requests.

- **Creating a new dynamic cluster on Linux using WebSphere Virtual Enterprise**
  After installing and configuring IBM WebSphere Portal on your primary node and all additional nodes, you can create a new dynamic cluster. A dynamic cluster monitors performance and load information and is able to dynamically create and remove cluster members based on the workload.

**Parent topic:** Setting up a cluster on Linux

**Previous topic:** Preparing the primary node on Linux

**Next topic:** Preparing a remote Web server when portal is installed on Linux
Creating a static cluster on Linux

After installing IBM® WebSphere® Portal on the primary node, configuring a remote database, and preparing the primary node to communicate with the Deployment Manager, you can create your static cluster to handle failover requests.

Perform the following steps to create the cluster:

1. If the Deployment Manager is configured to use a stand-alone LDAP user registry, update the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory, on the primary node with the stand-alone LDAP user registry property values from the Deployment Manager. You can find these settings under the VMM Stand-alone LDAP configuration heading.

   **Important:** Ensure that you set WasUserId and WasPassword to the Deployment Manager user ID and password.

2. Stop the server1 and WebSphere_Portal servers on the primary node and ensure that the following parameters are set correctly in the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory:

   **Note:** Although you can add these parameters directly to any task that you run while creating your cluster, you may want to add them to the properties file while creating your cluster and then remove them when you are finished to keep your environment secure.

   A. Set WasSoapPort to the port used to connect remotely to the deployment manager.
   B. Set WasRemoteHostName to the full host name of the server used to remotely connect to the deployment manager.
   C. Verify that WasPassword is set to your deployment manager password.
   D. Verify that PortalAdminPwd is set to your WebSphere Portal password.
   E. Verify thatClusterName is set.
   F. Verify that PrimaryNode is set to true.

3. Make a backup copy of the wp_profile_root/config/cells/cell_name/wim/config/wimconfig.xml and wp_profile_root/config/cells/cell_name/wim/model/wimxmlextension.xml, if available, files.

4. Run the ./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=dmgr_password task, from the wp_profile_root/ConfigEngine directory of the primary node.

   **Note:** If you want to specify custom ports for the nodeagent, add the -DPortPropsFile=full to portsfile parameter to the cluster-node-config-pre-federation task. You can use the ports files that are found on the Setup CD for WebSphere_Portal and server1 as a guide.

   **Note:** You may receive a message about accepting an SSL signer certificate. Failure to accept the SSL signer certificate will cause the script to fail. Alternatively, the com.ibm.ssl.enableSignerExchangePrompt flag can be enabled in the ssl.client.props file for “DefaultSSLSettings” in order to allow acceptance of the signer during the connection attempt.

   **Warning:** If the cluster-node-config-pre-federation fails for any reason, you must perform the following steps before rerunning the task:

   A. Remove the node if the AddNode task succeeded.
   B. Log on to the deployment manager and perform the following steps if the items exist:
      1. Remove all enterprise applications.
      2. Remove the WebSphere_Portal server definition.
      3. Remove the WebSphere Portal JDBC Provider.

   **Note:** If you are migrating from a version prior to Version 6.1.0 that contains Lookaside data, read the following technote: Migrating with Lookaside Data.
Optional: If you configured a database user registry or a property extension (lookaside) database, run the following task to set up access to the database drivers:

**Note:** You will also need to run this task if you migrated the primary node from a release prior to Version 6.1.0, even if you did not configure a database user registry or a property extension (lookaside) database.

A. Set the property value for `federated.db.DbType` if using a database user registry or set the property value for `laDbType` if using a property extension database in the `wkplc.properties` file.
   
   **Note:** If you migrated the primary node from a version prior to Version 6.1.0 and you are not using a database user registry or a property extension database, set the property value for `federated.db DbType` to the same value that is in the `wkplc.properties` file on the primary node.

B. Run the `./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=dmgr_password -DDbDomain=la|federated.db -DvmmNodeName=node_name -Db_type.NodeDbLibrary=local full path of the database jars` task from the `wp_profile_root/ConfigEngine` directory to create the variable used to access the VMM database jars.

   **Note:** `VmmNodeName` is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The `db_type` in `db_type.NodeDbLibrary` should be set to the type of database you are using, for example `db2`.

6. Run the `./ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=dmgr_password` task.

7. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell's user registry. Run the `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task, from the `wp_profile_root/ConfigEngine` directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

   **Attention:** The password value for `-DWasPassword` is the Deployment Manager administrative password.

   **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

   **Fast path:** If the value for `newAdminGroupId` contains a space; for example `Software Group`, open the `wkplc.properties` file and add the values for `newAdminId`, `newAdminPw`, and `newAdminGroupId`. Save your changes and run the `ConfigEngine.bat wp-change-portal-admin-user -DWasPassword=dmgr_password` task.

   **Important:** If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user` task until after the `cluster-node-config-cluster-setup` (static cluster) or `cluster-node-config-dynamic-cluster-setup` (dynamic cluster) task completes. After running the `wp-change-portal-admin-user` task, start or restart the WebSphere_Portal server to use the updated administrator user ID.

   **Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the `wp-change-portal-admin-user` task. `-DnewAdminPw` is an optional parameter to update the Administrative password in the `wkplc.properties` file if required.

8. Run the `./ConfigEngine.sh cluster-node-config-cluster-setup -DWasPassword=dmgr_password` task.

9. Configure the cluster to use an external Web server to take advantage of features such as workload management. Choose one of the following options:

   - Configuring a Web server and an application server on separate machines (remote)
   - Configuring a Web server and an application server profile on the same machine
   - Configuring a Web server and a deployment manager profile on the same machine

   **Note:** Start with the step about launching the Plug-ins installation wizard.

10. Perform the following steps to access the Web Content Management content through an external Web server:

    A. Log on to the deployment manager administrative console.
B. Select Environment > WebSphere Variables.

C. From the Scope drop-down menu, select the Node=nodename, Server=servername option to narrow the scope of the listed variables, where Node=nodename is the node that contains the application server.

D. Update the WCM_HOST variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.

E. Update the WCM_PORT variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.

F. Perform the following steps to synchronize the node with the deployment manager:
   1. Select System Administration > Nodes.
   2. Select the node that you want to synchronize from the list.
   3. Click Full Resynchronize.

G. Log off of the deployment manager administrative console.

11. Run the following tasks to propagate your changes:Note: WebSphere_Portal_nodename is the name of the node’s WebSphere Portal server; but if you customized the server name, the default name changes. If you are uncertain of the server name, run the serverstatus -all task to get a list of the server names and their status.

   A. ./stopManager.sh -username admin_userid -password admin_password, from the dmgr_profile_root\bin directory

   B. ./stopNode.sh -username admin_userid -password admin_password, from the wp_profile_root/bin directory

   C. ./stopServer.sh WebSphere_Portal_nodename -username admin_userid -password admin_password

   D. ./startManager.sh, from the dmgr_profile_root\bin directory

   E. ./startNode.sh, from the wp_profile_root/bin directory

   F. ./startServer.sh WebSphere_Portal_nodename

12. Change the task list settings to point to the cluster instead of a single server if using Portal business process integration. See one of the following sections in the Information Center for information:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross cell</td>
<td>Adding a BPI-enabled portal server to a managed cell in a cross-cell setup</td>
</tr>
<tr>
<td>Single cell</td>
<td>Adding a BPI-enabled portal server to a managed cell in a single-cell setup</td>
</tr>
</tbody>
</table>

**Parent topic:** Choosing the type of cluster to create on Linux

**Related tasks**

- Adding a BPI-enabled portal server to a managed cell in a cross-cell setup
- Adding a BPI-enabled portal server to a managed cell in a single-cell setup
Creating a new dynamic cluster on Linux using WebSphere Virtual Enterprise

After installing and configuring IBM® WebSphere® Portal on your primary node and all additional nodes, you can create a new dynamic cluster. A dynamic cluster monitors performance and load information and is able to dynamically create and remove cluster members based on the workload.

Before creating your dynamic cluster, install and configure the Deployment Manager and perform all the tasks under “Preparing the primary node.” On the Deployment Manager system, install WebSphere Virtual Enterprise and augment the deployment manager profile. On all WebSphere Portal nodes, install WebSphere Virtual Enterprise and augment the wp_profile profile. See Installing and configuring the product for information about installing WebSphere Virtual Enterprise. See Augmenting profiles for information about augmenting the Deployment Manager and wp_profile profiles.

- **Prerequisites**
  - PK97393 for all WebSphere Application Server Versions prior to 6.1.0.5
  - Installing and configuring the product:
    - Augmenting profiles

Perform the following steps to create a dynamic cluster:

1. Perform the following steps for on the primary node of the dynamic cluster:
   A. Prepare the node for the dynamic cluster; perform all tasks under Preparing the primary node.
   B. If the Deployment Manager is configured to use a stand-alone LDAP user registry, update the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory, on the primary node with the stand-alone LDAP user registry property values from the Deployment Manager. You can find these settings under the VMM Stand-alone LDAP configuration heading.
   C. Stop the server1 and WebSphere_Portal servers on the primary node and ensure that the following parameters are set correctly in the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory:
      1. Set WasSoapPort to the port used to connect remotely to the deployment manager.
      2. Set WasRemoteHostName to the full host name of the server used to remotely connect to the deployment manager.
      3. Verify that WasPassword is set to your deployment manager password.
      4. Verify that PortalAdminPwd is set to your WebSphere Portal password.
      5. Verify that ClusterName is set.
      6. Verify that PrimaryNode is set to true.
   D. Run the ./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=dmgr_password task, from the wp_profile_root/ConfigEngine directory of the primary node.

   **Warning:** If the cluster-node-config-pre-federation fails for any reason, you must perform the following steps before rerunning the task:
   1. Remove the node if the AddNode task succeeded.
   2. Log on to the deployment manager and perform the following steps if the items exist:
      A. Remove all enterprise applications.
      B. Remove the WebSphere_Portal server definition.
      C. Remove the WebSphere Portal JDBC Provider.

   **Note:** If you are migrating from a version prior to Version 6.1.0 that contains Lookaside data, read the following
technote: Migrating with Lookaside Data.

E. Run the following task. Include each node name as a comma separated list in the command:

1. Set the property value for `federated.db.DbType` if using a database user registry or set the property value for `la.DbType` if using a property extension database in the wkplc.properties file. **Note:** If you migrated the primary node from a version prior to Version 6.1.0 and you are not using a database user registry or a property extension database, set the property value for `federated.db.DbType` to the same value that is in the wkplc.properties file on the primary node.

2. Run the `./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=dmgr_password -DDomain=la|federated.db -DVmmNodeName=node_name -Db_type.NodeDbLibrary=local full path of the database jars` task from the `wp_profile_root/ConfigEngine` directory to create the variable used to access the VMM database jars. **Note:** `VmmNodeName` is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The `db_type` in `db_type.NodeDbLibrary` should be set to the type of database you are using, for example `db2`.

F. Run the `./ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=dmgr_password` task.

G. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell's user registry. Run the `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=dmgr_password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task from the `wp_profile_root/ConfigEngine` directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

**Important:** If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user` task until after the `cluster-node-config-dynamic-cluster-setup` task completes.

After running the `wp-change-portal-admin-user` task, start or restart the WebSphere Portal server to use the updated administrator user ID.

**Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager cell before running the `wp-change-portal-admin-user` task. `-DnewAdminPw` is an optional parameter to update the Administrative password in the `wkplc.properties` file if required.

2. Log on to the deployment manager administrative console.

3. Perform the following steps to create a node group:
   A. Click **System administration > Node groups**.
   B. Click **New**.
   C. Type the node group **Name**.
   D. Optional: Type any information about the node group in the **Description** text box.
   E. Click **OK**.
   F. Click the **Save** link to save your changes to the master configuration.

4. Perform the following steps to add members to the node group:
   A. Click **System administration > Node groups**.
   B. Click on the name of the node group that you want to add members to.
   C. Click **Node group members** under Additional Properties.
   D. Click **Add**.
   E. Select the primary node and then click **Add**.
   F. Click the **Save** link to save your changes to the master configuration.

5. Perform the following steps to create a dynamic cluster in the node group:
   A. Click **Servers > Dynamic clusters**.
   B. Click **New**.
C. Select WebSphere Application Server from the Server Type pull-down menu and then click Next.
D. Type the cluster name in the Dynamic cluster name text box and then click Next. Type the same value that you provided for the ClusterName parameter in the wkplc.properties file of your primary node.
E. Remove all default membership policies and then click Subexpression builder.
F. Enter the following information in the Subexpression builder window:
   1. Select and from the Logical operator pull-down menu.
   2. Select Nodegroup from the Select operand pull-down menu.
   3. Select Equals (=) from the Operator pull-down menu.
   4. Type the nodegroup name you created in the previous step in the Value text box.
   5. Click Generate subexpression.
   6. Click Append.
G. Click Preview membership to verify that all nodes included in the nodegroup display and then click Next.
H. Click the Create the cluster member using an existing server as a template radio button and then select the WebSphere_Portal server for the primary node from the pull-down menu.
I. Click Next.
J. Specify the dynamic cluster properties and then click Next.
K. Review the summary page to verify your actions and then click Finish.
L. Click the Save link to save your changes to the master configuration.

6. Define or verify the following parameters in the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory:
   A. Set ClusterName to the name of the new dynamic cluster.
   B. Verify that CellName is set to the deployment manager cell.
   C. Verify that NodeName is set to the local WebSphere Portal node.
   D. Set ServerName to the server that will be used for the dynamic cluster member on this node.
      Note: Log on to the deployment manager administrative console and click Dynamic Clusters > PortalCluster > Dynamic cluster members to find the name of the server used for the dynamic cluster.
   E. Verify that PrimaryNode is set to true.

7. Run the ./ConfigEngine.sh cluster-node-config-dynamic-cluster-setup -DWasPassword=dmgr_password task from the wp_profile_root/ConfigEngine directory to create the dynamic cluster.
   Note: This task may display a warning message since the cluster already exists but it will proceed to create the new dynamic cluster; therefore, the warning can be ignored.

8. Perform the following steps to access the Web Content Management content through an external Web server:
   A. Log on to the deployment manager administrative console.
   B. Select Environment > WebSphere Variables.
   C. From the Scope drop-down menu, select the Node-nodename, Server-servername option to narrow the scope of the listed variables, where Node-nodename is the node that contains the application server.
   D. Update the WCM_HOST variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.
   E. Update the WCM_PORT variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.
   F. Perform the following steps to synchronize the node with the deployment manager:
      1. Select System Administration > Nodes.
      2. Select the node that you want to synchronize from the list.
      3. Click Full Reresynchronize.
   G. Log off of the deployment manager administrative console.

9. Run the following tasks to propagate your changes:
   Note: WebSphere_Portal_nodename is the name of the node's WebSphere Portal server; but if you customized the server name, the default name changes. If you are uncertain of the
server name, run the `serverstatus -all` task to get a list of the server names and their status.

A. `.stopManager.sh -username admin_userid -password admin_password`, from the `dmgr_profile_root\bin` directory

B. `.stopNode.sh -username admin_userid -password admin_password`, from the `wp_profile_root/bin` directory

C. `.stopServer.sh WebSphere_Portal_nodename -username admin_userid -password admin_password`

D. `.startManager.sh`, from the `dmgr_profile_root\bin` directory

E. `.startNode.sh`, from the `wp_profile_root/bin` directory

F. `.startServer.sh WebSphere_Portal_nodename`

**Parent topic:** Choosing the type of cluster to create on Linux
Preparing a remote Web server when portal is installed on Linux

Install and configure the Web server plug-in, provided by IBM® WebSphere® Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

- **Prerequisites**
  - Preparing the WebSphere Application Server Deployment Manager on Linux
  - Preparing prerequisite and corequisite software on Linux
  - Preparing your Linux operating system
  - Preparing the primary node on Linux

Perform the following steps to install and configure your Web server:

1. Install and configure the Web server; refer to the Web server documentation for information.
2. If using IBM Lotus® Domino®, edit the `NOTES.INI` file on the Web server. Set the `HTTPEnableConnectorHeaders` parameter to 1.
3. If using IBM HTTP Server or Apache Server, edit the `httpd.conf` file on the Web server. Set the `AllowEncodedSlashes` directive to on; the directive should be added at the root level as a global directive.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>See the appropriate HTTP Server documentation</td>
<td>IBM HTTP Server</td>
</tr>
<tr>
<td>See the appropriate Apache Server documentation</td>
<td>AllowEncodedSlashes directives</td>
</tr>
</tbody>
</table>

4. Stop the Web server.
5. Install and configure the Web server plug-in on the system where the Web server is located using the Plug-ins installation wizard provided with WebSphere Application Server. Refer to Selecting a Web server topology diagram and roadmap for information. **Note:** The WebSphere Application Server Web server instructions are for the default profile. If not using the default profile, refer to Configuring a Web server for a non-default profile.

**If using WebDAV:** After successfully installing the Web server plug-in, locate and open your `plugin-cfg.xml` file and set `AcceptAllContent` to true.

**Important:** Depending on how you use the Web server, you may need to adjust the `ServerIOTimeout` value, which defines how long the plug-in should wait for a response from the application. The recommended minimum value is 60 but you may need to adjust this value higher if you are retrieving data from a database. To update this value, locate and open your `plugin-cfg.xml` file and set `ServerIOTimeout` to a value that is appropriate for your business needs. For additional information, see Common questions about the Web server plug-in.

6. If you are using Sun Web Server Version 7 update 8, read Technote 1448262 and perform the steps to resolve the HTTP 408/409 error.
7. Start the Web server.
Configuring WebSphere Portal to use a user registry on Linux in a clustered environment

Configure user registry security on IBM® WebSphere® Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

Prior to configuring security, you should use the IBM WebSphere Application Server backupConfig task to create and store a backup of the IBM WebSphere Portal configuration; see backupConfig command for information.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on Linux
  - Preparing your Linux operating system
  - Preparing the primary node on Linux

Perform the following tasks to configure WebSphere Portal to use a user registry:

- **Preparing user registries on Linux**
  Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

- **Choosing your user registry model on Linux in a clustered environment**
  Choose between securing IBM WebSphere Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

**Parent topic:** Setting up a cluster on Linux

**Previous topic:** Preparing a remote Web server when portal is installed on Linux

**Next topic:** Preparing additional nodes on Linux
Preparing user registries on Linux

Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

Choose the appropriate LDAP server to install and setup:

- **Preparing a Domino Directory server**
  If you plan to use a Domino Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a SecureWay Security Server**
  If you plan to use a SecureWay Security Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Novell eDirectory**
  If you plan to use a Novell eDirectory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Sun Java System Directory Server**
  If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Tivoli Directory Server**
  If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

**Parent topic:** Configuring WebSphere Portal to use a user registry on Linux in a clustered environment
Preparing a Domino Directory server

If you plan to use a Domino Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Domino Directory:

1. Perform the following steps to install the Domino Directory:
   A. Go to Lotus Domino documentation.
   B. Select the appropriate version tab for your product.
   C. Click the Download/View online link for the Lotus Domino for multiple platforms Information Center.
   D. Click Domino Administrator Help > Installation > Installing and setting up Domino servers > Server installation > Installing Domino on Linux and perform this task.
   E. Click Domino Administrator Help > Installation > Installing and setting up Domino servers > The Domino server setup program and perform this task.

2. Perform the following steps as a guide to create the WebSphere Portal administrative user:
   A. Navigate to the People view of the Domino Directory and then click Add Person.
   B. Enter the following values in the New Person form to create the wpsbind user:
      - Last Name
        - wpsbind
      - User Name
        - wpsbind/DominoDomain, where DominoDomain is your Lotus Domino Internet domain
        - Note: Make sure you enter two values in the User Name field, where the first value includes the Lotus Domino domain.
      - Short name/UserID
        - wpsbind
      - Internet password
        - wpsbind
   C. Click Save and Close to save the new person record for wpsbind and return to the People view.
   D. Click Add Person and enter the following values in the New Person form to create the wpsadmin user:
      - Last Name
        - wpsadmin, where wpsadmin in the user ID for the WebSphere Portal Administrator
      - User Name
        - wpsadmin/DominoDomain, where DominoDomain is your Lotus Domino Internet domain
        - Note: Make sure you enter two values in the User Name field, where the first value includes the Lotus Domino domain.
      - Short name/UserID
        - wpsadmin
      - Internet password
        - wpsadmin
   E. Click Save and Close to save the new person record for wpsadmin and return to the People view.
   F. Navigate to the Groups view and click Add Group.
   G. Enter the following values in the New Group form on the Basic tab:
      - Group name
Note: If you plan to configure WebSphere Portal for multiple user registries and your Lotus Domino LDAP will share a realm with another user registry, you must use the hierarchical naming convention for the group names, for example: wpsadmins/DominoDomain, to avoid unexpected results during WebSphere Portal runtime.

- **Group type**
  - Multi-purpose

- **Members**
  - wpsbind/DominoDomain
  - wpsadmin/DominoDomain

  **Note:** You can add additional administrator users if required.

H. Click **Save and Close** to save the wpsadmins group with the wpsbind and wpsadmin users as members.

3. Perform the following steps to update the access control list for the Domino Directory:
   A. Open the names.nsf file in the Lotus Domino Administrator or Lotus Notes client.
   B. Click **File > Application > Access Control** from the main menu to open the access control list for the file.
   C. In the **Access Control List > Basics** panel, ensure that the wpsadmins group has either Author or Editor access.
   D. Add the following Role Types to the wpsadmins group:
      - GroupCreator
      - GroupModifier
      - UserCreator
      - UserModifier
   E. Click **OK**.

**Parent topic:** Preparing user registries on Linux
Preparing a SecureWay Security Server

If you plan to use a SecureWay Security Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare SecureWay Security Server:

1. Install SecureWay Security Server; refer to the IBM SecureWay™ Security Server for z/OS® and OS/390® for information.

2. Perform the following steps to create the WebSphere Portal administrative user:
   
   A. Optional: Perform the following steps to create a new directory suffix:
      
      1. Click the **Server Administration** folder, located in the left-hand navigation of the directory server console.
      2. Click the **Manage Server Properties** folder under the Server Administration folder and then select **Suffixes** on the main page.
      3. Type the **Base DN** name for the suffix; for example: \texttt{dc=yourcompany,dc=com}.
      4. Click **Add**.
      5. Click **OK** to save your changes.
      6. Stop and restart the LDAP server.

   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      
      - Use the **PortalUsers.ldif** file as a working example and adapted appropriately to work with your LDAP server.
      - Use the **ContentUsers.ldif** file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.

   C. Replace every \texttt{dc=yourco,dc=com} with your suffix.

   D. Replace any prefixes and suffixes that are unique to your LDAP server.

   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.

   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the **objectclasses** to **accessGroup**. If using Tivoli Access Manager Version 6, set the **objectclasses** to **groupOfNames**.

   G. Save your changes.

   H. Follow the instructions provided with your directory server to import the LDIF file.

   I. Stop and restart the LDAP server.

**Parent topic:** Preparing user registries on Linux
Preparing a Novell eDirectory

If you plan to use a Novell eDirectory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Novell eDirectory:

1. Install Novell eDirectory; refer to eDirectory for information.
2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every dc=yourco,dc=com with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.
   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on Linux
Preparing a Sun Java System Directory Server

If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Sun Java System Directory Server:

1. Install Sun Java System Directory Server; refer to Oracle software for information.

2. Perform the following steps to create the WebSphere Portal administrative user:

   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the **Server Administration** folder, located in the left-hand navigation of the directory server console.
      2. Click the **Manage Server Properties** folder under the Server Administration folder and then select **Suffixes** on the main page.
      3. Type the **Base DN** name for the suffix; for example: `dc=yourcompany,dc=com`.
      4. Click **Add**.
      5. Click **OK** to save your changes.
      6. Stop and restart the LDAP server.

   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the **PortalUsers.ldif** file as a working example and adapted appropriately to work with your LDAP server.
      - Use the **ContentUsers.ldif** file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
      C. Replace every `dc=yourco,dc=com` with your suffix.
      D. Replace any prefixes and suffixes that are unique to your LDAP server.
      E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
      F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the **objectclasses** to **accessGroup**. If using Tivoli Access Manager Version 6, set the **objectclasses** to **groupOfNames**.
      G. Save your changes.
      H. Follow the instructions provided with your directory server to import the LDIF file.
      I. Stop and restart the LDAP server.

**Parent topic:** Preparing user registries on Linux
Preparing a Tivoli Directory Server

If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Tivoli Directory Server:

   
   **Note:** Due to a restriction in Tivoli Directory Server, users or groups must not contain a Turkish uppercase dotted I or lowercase dotted i in the DN as this will prevent correct retrieval of that user or group.

2. Perform the following steps to create the WebSphere Portal administrative user:
   
   A. Optional: Perform the following steps to create a new directory suffix:
      
      1. Click the **Server Administration** folder, located in the left-hand navigation of the directory server console.
      2. Click the **Manage Server Properties** folder under the Server Administration folder and then select **Suffixes** on the main page.
      3. Type the **Base DN** name for the suffix; for example: `dc=yourcompany,dc=com`.
      4. Click **Add**.
      5. Click **OK** to save your changes.
      6. Stop and restart the LDAP server.

   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      
      - Use the `PortalUsers.ldif` file as a working example and adapted appropriately to work with your LDAP server.
      - Use the `ContentUsers.ldif` file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.

   C. Replace every `dc=yourco,dc=com` with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.

   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the `objectclasses` to `accessGroup`. If using Tivoli Access Manager Version 6, set the `objectclasses` to `groupOfNames`.

   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.

**Parent topic:** Preparing user registries on Linux
Choosing your user registry model on Linux in a clustered environment

Choose between securing IBM® WebSphere® Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

Choose one of the following user registry models:

- **Configuring a stand-alone LDAP user registry on Linux in a clustered environment**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Configuring the default federated repository on Linux in a clustered environment**
  Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

- **Adapting the attribute configuration**
  After installing IBM WebSphere Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

Parent topic: Configuring WebSphere Portal to use a user registry on Linux in a clustered environment
Configuring a stand-alone LDAP user registry on Linux in a clustered environment

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Configuring a stand-alone LDAP user registry on Linux in a clustered environment**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry to store all user account information for authorization.

- **Configuring a stand-alone LDAP user registry over SSL on Linux in a clustered environment**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

**Parent topic:** Choosing your user registry model on Linux in a clustered environment
Configuring a stand-alone LDAP user registry on Linux in a clustered environment

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry to store all user account information for authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps. If you need to rerun the wp-modify-ldap-security task to change the LDAP repositories or because the task failed, you must choose a new name for the realm using the standalone.ldap.realm parameter or you can set ignoreDuplicateIDs=true in the wkplc.properties file, before rerunning the task.

Perform the following steps to configure a standalone LDAP user registry:

Note: Use the wp_security_xxx.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_security_xxx.properties helper file.

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

2. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM Standalone LDAP configuration heading:

   - standalone.ldap.id
   - standalone.ldap.host
   - standalone.ldap.port
   - standalone.ldap.blindDN
   - standalone.ldap.bindPassword
   - standalone.ldap.idapServerType
   - standalone.ldap.userIdMap
   - standalone.ldap.groupIdMap
   - standalone.ldap.groupMemberIdMap
   - standalone.ldap.userFilter
   - standalone.ldap.groupFilter
   - standalone.ldap.serverId
   - standalone.ldap.serverPassword
   - standalone.ldap.realm
   - standalone.ldap.primaryAdminId
   - standalone.ldap.primaryAdminPassword
   - standalone.ldap.primaryPortalAdminId
   - standalone.ldap.primaryPortalAdminPassword
   - standalone.ldap.primaryPortalAdminGroup
   - standalone.ldap.baseDN
3. Required: Enter a value for the following required entity types parameters in the wkplc.properties file under the LDAP entity types heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.idap.et.group.objectClasses
   - standalone.idap.et.group.objectClassesForCreate
   - standalone.idap.et.group.searchBases
   - standalone.idap.et.personaccount.objectClasses
   - standalone.idap.et.personaccount.objectClassesForCreate
   - standalone.idap.et.personaccount.searchBases

4. Required: Enter a value for the following required group member parameters in the wkplc.properties file under the Group member attributes heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.idap.gm.groupMemberName
   - standalone.idap.gm.objectClass
   - standalone.idap.gm.scope
   - standalone.idap.gm.dummyMember

5. Required: Enter a value for the following required relative distinguished name (RDN®) parameters in the wkplc.properties file under the Default parent, RDN attribute heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - standalone.idap.personAccountParent
   - standalone.idap.groupParent
   - standalone.idap.personAccountRdnProperties
   - standalone.idap.groupRdnProperties

6. Save your changes to the wkplc.properties file.

7. Optional: Enter the following Web content authors parameters in the wkplc_comp.properties file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. Note: See the wkplc_comp.properties file for specific information about the required parameters and for advanced parameters.
   - WcmContentAuthorsGroupId
   - WcmContentAuthorsGroupCN

8. Run the ./ConfigEngine.sh validate-standalone-ldap -DWasPassword=password task to validate your LDAP server settings. Attention: If you have not deleted the default file repository, WasPassword is the value entered during installation and not a value found in your LDAP user registry.
   Note: During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press y then Enter.

9. Run the ./ConfigEngine.sh wp-modify-ldap-security -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to set the stand-alone LDAP user registry.

10. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

11. Run the ./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to check that all defined attributes are available in the configured LDAP user registry. Important: When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

12. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the
LDAP directory.

**Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

A. **Edit the** `wp_profile_root` /PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties file.

B. **Add the following lines to the file:**

```plaintext
uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
```

Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. **Save your changes and close the file.**

D. **Run the** `./ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWenPassword=password` task, located in the `wp_profile_root`/ConfigEngine directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: *Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management*

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

13. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring a stand-alone LDAP user registry on Linux in a clustered environment

**Related tasks**
- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation
Configuring a stand-alone LDAP user registry over SSL on Linux in a clustered environment

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to configure a standalone LDAP user registry over SSL:

Using the wp_security_xxx.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_security_xxx.properties helper file.

1. To specify the LDAP server's SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:

   • **Server trust store**
     
     A. Add the certificate to the trust store:
     
        1. Log in to the WebSphere Application Server Administrative Console.
        2. Navigate to Security > SSL certificate and key management > SSL configurations.
        3. Click the appropriate SSL configuration from the list. For example,
           - Stand-alone environments: NodeDefaultSSLSettings
           - Clustered environments: CellDefaultSSLSettings
        
           **Clustered environments:** Ensure the setting for SSL configuration for outbound connection matches your SSL settings.
        4. Click Key stores and certificates.
        5. Click the appropriate trust store from the list; for example, NodeDefaultTrustStore.
        6. Click Signer certificates, click Add, and then enter the following information:
           - Type the Alias the key store uses for the signer certificate.
           - Type the File name where the signer certificate is located.
        7. Click OK and then click Save to save the changes to the master configuration.

   B. Retrieve the certificate from the port:
     
        1. Log in to the WebSphere Application Server Administrative Console.
        2. Navigate to Security > SSL certificate and key management > SSL configurations.
        3. Click the appropriate SSL configuration from the list. For example,
           - Stand-alone environments: NodeDefaultSSLSettings
Clustered environments: CellDefaultSSLSettings

Clustered environments: Ensure the setting for SSL configuration for outbound connection matches your SSL settings.

4. Click Key stores and certificates.

5. Click the appropriate trust store from the list; for example, NodeDefaultTrustStore.

6. Click Signer certificates, click Retrieve from port, and then enter the following information:
   - Type the Host name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the SSL Port used when attempting to retrieve the signer certificate.
   - Type the Alias the key store uses for the signer certificate.

7. Click Retrieve signer information to retrieve the certificate from the port.

8. Click OK and then click Save to save the changes to the master configuration.

Client trust store Note: This task might report an error, but it does successfully update the trust store. You can ignore the error message.

A. See Secure installation for client signer retrieval.

B. Run the retrieveSigners task from the wp_profile_root/bin directory; see retrieveSigners command for information. In a deployed environment, you will need to run the retrieveSigners task, for any federated node, against the Deployment Manager. Note: This task might report an error, but it does successfully update the trust store. You can ignore the error message.

C. Update the trust store properties file:
   1. Use a text editor to open the ssl.client.props file, located in the wp_profile_root/properties directory.
   2. Change the com.ibm.ssl.trustStore parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter

```
com.ibm.ssl.trustStore=${CONFIG_ROOT}\cells\wpsbvt\nodes\wpsbvt\trust.p12
```

   to use the default trust store.

   3. Save your changes.

2. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

3. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM Stand-alone LDAP configuration heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.

   - standalone.idap.id
   - standalone.idap.host
   - standalone.idap.port
   - standalone.idap.blindDN
   - standalone.idap.bindPassword
   - standalone.idap.idapServerType
   - standalone.idap.userldMap
   - standalone.idap.groupldMap
   - standalone.idap.groupMemberldMap
   - standalone.idap.userFilter
   - standalone.idap.groupFilter
   - standalone.idap.serverId
   - standalone.idap.serverPassword
   - standalone.idap.realm
   - standalone.idap.primaryAdminId
   - standalone.idap.primaryAdminPassword
   - standalone.idap.primaryPortalAdminId
4. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading. **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.ldap.et.group.objectClasses`
   - `standalone.ldap.et.group.objectClassesForCreate`
   - `standalone.ldap.et.group.searchBases`
   - `standalone.ldap.et.personaccount.objectClasses`
   - `standalone.ldap.et.personaccount.objectClassesForCreate`
   - `standalone.ldap.et.personaccount.searchBases`

5. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attributes heading. **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.ldap.gm.groupMemberName`
   - `standalone.ldap.gm.objectClass`
   - `standalone.ldap.gm.scope`
   - `standalone.ldap.gm.dummyMember`

6. Required: Enter a value for the following required relative distinguished name (RDN) parameters in the `wkplc.properties` file under the Default parent, RDN attribute heading. **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.ldap.personAccountParent`
   - `standalone.ldap.groupParent`
   - `standalone.ldap.personAccountRdnProperties`
   - `standalone.ldap.groupRdnProperties`

7. Enter a value for the following parameters to enable Secure Socket Layers (SSL). **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   Required parameters:
   - `standalone.ldap.sslEnabled`
   - `standalone.ldap.sslConfiguration`

   Optional parameters:
   - `standalone.ldap.certificateMapMode`
   - `standalone.ldap.certificateFilter`

8. Save your changes to the `wkplc.properties` file.

9. Run the `./ConfigEngine.sh validate-standalone-ldap -DWasPassword=password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, `WasPassword` is the value entered during installation and not a value found in your LDAP user registry. **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press `y` then `Enter`.

10. Run the `./ConfigEngine.sh wp-modify-ldap-security -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to set the stand-alone LDAP user registry.

11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: **Starting and stopping servers, deployment managers, and node agents.**

12. Run the `./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured...
LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring a stand-alone LDAP user registry on Linux in a clustered environment

**Related tasks**
- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation
Configuring the default federated repository on Linux in a clustered environment

Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

Perform the following tasks to add user registries to the default federated repository:

- **Configuring a federated LDAP user registry on Linux in a clustered environment**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Adding a database user registry on Linux in a clustered environment**
  Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

- **Adding realm support on Linux in a clustered environment**
  A realm is a group of users from one or more user registries that form a coherent group within IBM WebSphere Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

**Parent topic:** Choosing your user registry model on Linux in a clustered environment
Configuring a federated LDAP user registry on Linux in a clustered environment

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Adding an LDAP user registry on Linux in a clustered environment**
  Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time. If IBM Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=groupName` and the hierarchical format is `cn=groupName,o=root`. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as `wpsadmin`, this ID cannot exist in the LDAP you are adding.

- **Adding an LDAP user registry over SSL on Linux in a clustered environment**
  Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

Parent topic: Configuring the default federated repository on Linux in a clustered environment
Adding an LDAP user registry on Linux in a clustered environment

Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time. If IBM® Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=groupName` and the hierarchical format is `cn=groupName, o=root`. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as `wpsadmin`, this ID cannot exist in the LDAP you are adding.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

**Tip:** Perform these steps on the primary node only.

**Note:** Use the `wp_add_federated_xxx.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_add_federated_xxx.properties` helper file.

1. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

2. Required: Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated LDAP Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - `federated.ldap.id`
   - `federated.ldap.host`
   - `federated.ldap.port`
   - `federated.ldap.bindDN`
   - `federated.ldap.bindPassword`
   - `federated.ldap.ldapServerType`
   - `federated.ldap.baseDN`

3. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

   - `federated.ldap.et.group.objectClasses`
   - `federated.ldap.et.group.objectClassesForCreate`
   - `federated.ldap.et.group.searchBases`
   - `federated.ldap.et.personaccount.objectClasses`
4. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attribute heading: 
   
   - `federated.ldap.gm.groupMemberName`  
   - `federated.ldap.gm.objectClass`  
   - `federated.ldap.gm.scope`  
   - `federated.ldap.gm.dummyMember`  

   `Note:` See the properties file for specific information about the required parameters and for advanced parameters.

5. Save your changes to the `wkplc.properties` file.

6. Optional: Enter the following Web content authors parameters in the `wkplc_comp.properties` file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. 
   
   - `WcmContentAuthorsGroupId`  
   - `WcmContentAuthorsGroupCN`  

   `Note:` See the `wkplc_comp.properties` file for specific information about the required parameters and for advanced parameters.

7. Run the `./ConfigEngine.sh validate-federated-ldap -DWasPassword=password` task to validate your LDAP server settings. 

   `Attention:` If you have not deleted the default file repository, `WasPassword` is the value entered during installation and not a value found in your LDAP user registry.

   `Note:` During the validation task, you may receive the following prompt: `Add signer to the trust store now?`. Press `y` then `Enter`.

8. Run the `./ConfigEngine.sh wp-create-ldap -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to add an LDAP user registry to the default federated repository. 

   `Note:` Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: `Starting and stopping servers, deployment managers, and node agents`.

10. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:

    A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

    B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms:

    - `id`  
    - `baseDN`  
    - `nameInRepository`  

    `Note:` See the properties file for specific information about the required parameters and for advanced parameters.

    C. Save your changes to the `wkplc.properties` file.

    D. Run the `./ConfigEngine.sh wp-create-base-entry -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to create a base entry in a repository.

    E. Stop and restart all necessary servers to propagate your changes.

11. Optional: Run the `./ConfigEngine.sh wp-query-repository -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to list the names and types of configured repositories.

12. Run the `./ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured
LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

13. Perform the following steps to update the user registry where new users and groups are stored:

**Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

**Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

**Note:** See the properties file for specific information about the required parameters and for advanced parameters.

- `personAccountParent`
- `groupParent`
- `personAccountRdnProperties`
- `groupRdnProperties`

The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:

- `personAccountParent=dc=yourco,dc=com`
- `groupParent=dc=yourco,dc=com`

C. Save your changes to the `wkplc.properties` file.

D. Run the `./ConfigEngine.sh wp-set-entitytypes -DWasPassword=yourpassword` task, from the `wp_profile_root/ConfigEngine` directory, to delete the old attributes before adding the new attributes.

E. Stop and restart all necessary servers to propagate your changes.

14. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. Enter a value for `realmName` or leave blank to update the default realm.

C. Save your changes to the `wkplc.properties` file.

D. Run the `./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=yourpassword` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=yourpassword` task to disable the feature.

E. Stop and restart all necessary servers to propagate your changes.

15. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

**Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

A. Edit the `wp_profile_root/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties` file.

B. Add the following lines to the file:

```properties
uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN

cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
```
Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `./ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management.

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

16. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

17. Optional: **This step is required in a production environment.** Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

   **Important:**
   - Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
   - Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

   **Note:** If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

   A. Run the following task from the `wp_profile_root/ConfigEngine directory`: `./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword`. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` parameter.

   B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere Portal servers. In a standalone environment, restart the server1 and WebSphere Portal servers.

   C. Run the following task from the `wp_profile_root/ConfigEngine directory`: `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid`. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupid` parameters.

   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

   D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere Portal servers. In a standalone environment, restart the server1 and WebSphere Portal servers.
18. **Optional:** **This step is required in a production environment.** Remove the file system repository if you do not use it. The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: *Deleting the repository.*

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring a federated LDAP user registry on Linux in a clustered environment

**Related tasks**

- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation

**Related information**

- User IDs and passwords
Adding an LDAP user registry over SSL on Linux in a clustered environment

Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry over SSL to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

**Tip:** Perform these steps on the primary node only.

**Note:** Use the `wp_add_federated_xxx.properties` helper file, located in the `/wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_add_federated_xxx.properties` helper file.

1. To specify the LDAP server's SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:
   - **Server trust store**
     A. Add the certificate to the trust store:
       1. Log in to the WebSphere Application Server Administrative Console.
       2. Navigate to Security > SSL certificate and key management > SSL configurations.
       3. Click the appropriate SSL configuration from the list. For example,
         - Stand-alone environments: `NodeDefaultSSLSettings`
         - Clustered environments: `CellDefaultSSLSettings`

         **Clustered environments:** Ensure the setting for SSL configuration for outbound connection matches your SSL settings.

       4. Click **Key stores and certificates**.
       5. Click the appropriate trust store from the list; for example, `NodeDefaultTrustStore`.
       6. Click **Signer certificates**, click **Add**, and then enter the following information:
          - Type the Alias the key store uses for the signer certificate.
          - Type the File name where the signer certificate is located.
       7. Click **OK** and then click **Save** to save the changes to the master configuration.

   B. Retrieve the certificate from the port:
      1. Log in to the WebSphere Application Server Administrative Console.
2. Navigate to Security > SSL certificate and key management > SSL configurations.
3. Click the appropriate SSL configuration from the list. For example,
   - Stand-alone environments: NodeDefaultSSLSettings
   - Clustered environments: CellDefaultSSLSettings

   **Clustered environments:** Ensure the setting for SSL configuration for outbound connection matches your SSL settings.
4. Click Key stores and certificates.
5. Click the appropriate trust store from the list; for example, NodeDefaultTrustStore.
6. Click Signer certificates, click Retrieve from port, and then enter the following information:
   - Type the **Host** name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the SSL **Port** used when attempting to retrieve the signer certificate.
   - Type the **Alias** the key store uses for the signer certificate.
7. Click Retrieve signer information to retrieve the certificate from the port.
8. Click OK and then click Save to save the changes to the master configuration.

   **Client trust store Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.
   A. See Secure installation for client signer retrieval.
   B. Run the retrieveSigners task from the wp_profile_root/bin directory; see retrieveSigners command for information. In a deployed environment, you will need to run the retrieveSigners task, for any federated node, against the Deployment Manager. **Note:** This task might report an error, but it does successfully update the trust store. You can ignore the error message.
   C. Update the trust store properties file:
      1. Use a text editor to open the ssl.client.props file, located in the wp_profile_root/properties directory.
      2. Change the com.ibm.ssl.trustStore parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter com.ibm.ssl.trustStore=${CONFIG_ROOT}\cells\wpsbvt\nodes\wpsbvt\trust.p12 to use the default trust store.
      3. Save your changes.
   2. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

   3. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM Federated LDAP Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
      - federated.idap.id
      - federated.idap.host
      - federated.idap.port
      - federated.idap.blindDN
      - federated.idap.blindPassword
      - federated.idap.idapServerType
      - federated.idap.idap.baseDN

   4. Required: Enter a value for the following required entity types parameters in the wkplc.properties file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
      - federated.idap.et.group.objectClasses
      - federated.idap.et.group.objectClassesForCreate
      - federated.idap.et.group.searchBases
5. Required: Enter a value for the following required group member parameters in the wkplc.properties file under the Group member attribute heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
- federated.ldap.et.personaccount.objectClasses
- federated.ldap.et.personaccount.objectClassesForCreate
- federated.ldap.et.personaccount.searchBases

6. Enter a value for the following parameters to enable Secure Socket Layers (SSL): Note: See the properties file for specific information about the required parameters and for advanced parameters.
   Required parameters:
   - federated.ldap.gm.groupMemberName
   - federated.ldap.gm.objectClass
   - federated.ldap.gm.scope
   - federated.ldap.gm.dummyMember

   Optional parameters:
   - federated.ldap.certificateMapMode
   - federated.ldap.certificateFilter

7. Save your changes to the wkplc.properties file.

8. Run the ./ConfigEngine.sh validate-federated-ldap -DWasPassword=password task to validate your LDAP server settings. Attention: If you have not deleted the default file repository, WasPassword is the value entered during installation and not a value found in your LDAP user registry. Note: During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press y then Enter.

9. Run the ./ConfigEngine.sh wp-create-ldap -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add an LDAP user registry to the default federated repository. Note: Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

10. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

11. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:
   A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
   B. Enter a value for the following required parameters in the wkplc.properties file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms:
      Note: See the properties file for specific information about the required parameters and for advanced parameters.
      - id
      - baseDN
      - nameInRepository
   C. Save your changes to the wkplc.properties file.
   D. Run the ./ConfigEngine.sh wp-create-base-entry -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to create a base entry in a repository.
E. Stop and restart all necessary servers to propagate your changes.

12. Optional: Run the `./ConfigEngine.sh wp-query-repository -DWasPassword=password` task, from the `wp_profile_root`/ConfigEngine directory, to list the names and types of configured repositories.

13. Run the `./ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root`/ConfigEngine directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

14. Perform the following steps to update the user registry where new users and groups are stored:

   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root`/ConfigEngine/properties directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - `personAccountParent`
      - `groupParent`
      - `personAccountRdnProperties`
      - `groupRdnProperties`

      The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:

      - `personAccountParent=dc=yourco,dc=com`
      - `groupParent=dc=yourco,dc=com`

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root`/ConfigEngine directory, to delete the old attributes before adding the new attributes.

   E. Stop and restart all necessary servers to propagate your changes.

15. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root`/ConfigEngine/properties directory.

   B. Enter a value for `realmName` or leave blank to update the default realm.

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root`/ConfigEngine directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

   E. Stop and restart all necessary servers to propagate your changes.

16. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

   **Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` command.
task.

A. Edit the `wp_profile_root`/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties file.

B. Add the following lines to the file:

```
uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
```

Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `.ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password task`, located in the `wp_profile_root/ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standaloneldap.realm</code> in the wkplc.properties file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the wkplc.properties file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

17. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

18. Optional: **This step is required in a production environment.** Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

**Important:**
- Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
- Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

**Note:** If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

A. Run the following task from the `wp_profile_root/ConfigEngine` directory: `.ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword` **Important:** You must provide the full distinguished name (DN) for the `newAdminid` parameter.

B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

C. Run the following task from the `wp_profile_root/ConfigEngine` directory: `.ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` **Important:** You must provide the full distinguished name (DN) for the `newAdminid` and `newAdminGroupId` parameters.

**Additional parameter for stopped servers:** This task verifies the user against a running server server instance. If the
D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

19. Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it. The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: *Deleting the repository.*

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring a federated LDAP user registry on Linux in a clustered environment

**Related tasks**
- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation

**Related Information**
- User IDs and passwords
Adding a database user registry on Linux in a clustered environment

Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps. If you have WebSphere Application Server Version 7.0.x installed, you must install APAR PM23090 and APAR PM24181 for WebSphere Portal prior to running this task. Clusters with WebSphere Application Server Version 6.1.x do not require these APARs.

Perform the following steps to add a database user registry to the default federated repository; you must repeat these steps for each additional database user registry that you plan to add:

**Tip:** Perform these steps on the primary node only.

**Note:** Use the `wp_add_DB.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_add_DB.properties` helper file.

1. Set up a new database, including creating a new user with appropriate database privileges for accessing the database:

   **Instructions for setting up databases:** Refer to the appropriate documentation for the type of database you want to set up.

   **Consulting your database administrator:** The task of setting up a new database is typically performed by a database administrator. However, the following steps are provided for your reference in the event you are creating a stand-alone database for testing or demonstration purposes. Consult your database administrator before proceeding with the following steps if you plan to create a database for a production environment.

   **Table 1. Steps for creating a new database to use as a database user registry.**

<table>
<thead>
<tr>
<th>Database</th>
<th>Steps</th>
</tr>
</thead>
</table>

   1736
### DB2

Perform the following steps to create a DB2 database:

1. **Install DB2.**
2. **Enter the following database tuning commands:**
   ```
   CREATE DB dbname using codeset UTF-8
territory us PAGESIZE 8192
   
   UPDATE DB CFG FOR dbname USING
   appheapsz 4096
   
   UPDATE DB CFG FOR dbname USING
   app_ctl_heap_sz 1024
   
   UPDATE DB CFG FOR dbname USING
   stmtheap 32768
   
   UPDATE DB CFG FOR dbname USING
   dbheap 2400
   
   UPDATE DB CFG FOR dbname USING
   locklist 1000
   
   UPDATE DB CFG FOR dbname USING
   logfilsiz 4000
   
   UPDATE DB CFG FOR dbname USING
   logprimary 12
   
   UPDATE DB CFG FOR dbname USING
   logsecond 20
   
   UPDATE DB CFG FOR dbname USING
   logbufsz 32
   
   UPDATE DB CFG FOR dbname USING
   avg_appls 5
   
   UPDATE DB CFG FOR dbname USING
   locktimeout 30
   
   UPDATE DB CFG FOR dbname using
   AUTO_MAINT off
   ```

### Oracle

Perform the following steps to create an Oracle database:

1. **Install Oracle using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.**
2. Configure the database in Dedicated Server Mode.
3. Enter the recommended initial buffer pool sizes or set them according to your business needs:
   - `db_block_size = 8192`
   - `db_cache_size = 300M`
   - `db_files = 1024`
   - `log_buffer = 65536`
   - `open_cursors = 1500`
   - `pga_aggregate_target = 200M`
   - `pre_page_sga = true`
   - `processes = 300`
   - `shared_pool_size = 200M`

### SQL Server

Perform the following steps to create an SQL Server database:

1. **Install SQL Server.**
2. **Set Collation to case-sensitive.**
3. **Note:** Install SQL Server with the appropriate portal database collation so that your tempdb collation setting matches the collation you use for the property extension database. The tempdb collation is inherited from the master database, which you set when you install SQL Server.

2. Perform the following steps to define the **DbDriver** and **DbLibrary** parameter values:
   - Navigate to the following directory: `wp_profile_root/ConfigEngine/properties`
   - Locate and open `wkplc_dbtype.properties` with any text editor.
   - Enter a value for the following parameters under the appropriate database type properties heading:
     - `db_type.DbDriver`
     - `db_type.DbLibrary`
   - Save your changes.

3. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

4. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated Database Properties heading:**Note:** See the properties file for specific information about the required parameters and for advanced parameters.
1. Navigate to the following directory: `wp_profile_root/properties`
2. Locate and open `soap.client.props` with any text editor.
3. Locate the `com.ibm.SOAP.requestTimeout` parameter and change the value to 1000.
4. Save and close `soap.client.props`.

5. Perform the following steps in a clustered environment:
   A. Run the `./ConfigEngine.sh wp-prep-vmm-db-secured-environment -DWasPassword=password -DDbDomain=federated.db -Db_type.DmgrDbLibrary=local path of the database jars on the Deployment Manager -DDmgrNodeName=dmgr_node_name` task from the `wp_profile_root/ConfigEngine` directory to create the local Deployment Manager WebSphere variable used to access the database jars.
   
   **Note:** The `db_type` in `db_type.DmgrDbLibrary` should be set to the type of database you are using, for example `db2`. The **local full path of the database jars on the Deployment Manager** should be one of the following options:

   - `DB2 Type 2 driver`: `db2java.zip`
   - `DB2 Type 4 driver`: `db2jcc.jar;db2jcc_license_cu.jar`
   - `DB2 for z/OS Type 2 driver`: `db2java.zip`
   - `DB2 for z/OS Type 4 driver`: `db2jcc.jar;db2jcc_license_cisuz.jar;db2jcc_javax.jar`
   - `Oracle`: `ojdbc14.jar`
   - `SQL Server JDBC driver provided by Microsoft`: `sqljdbc.jar`
   - `SQL Server JDBC driver provided by DataDirect`: `sqlserver.jar;base.jar;util.jar`

   B. Run the following task. Include each node name as a comma separated list in the command:

   **Running the task:** You do not have to run this task more than once. You can run this task from any node in the cluster.

   1. Set the property value for `federated.db.DbType` in the `wkplc.properties` file if using a database user registry.
   2. Run the `./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=password -DDbDomain=federated.db -DVmmNodeName=node_name -Db_type.NodeDbLibrary=local full path of the database jars` task from the `wp_profile_root/ConfigEngine` directory on each node to create the variable used to access the VMM database jars.

   **Note:** `VmmNodeName` is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The `db_type` in `db_type.NodeDbLibrary` should be set to the type of database you are using, for example `db2`.

   C. Stop and restart all necessary servers to propagate your changes.

6. Stop and restart all necessary servers to propagate your changes.

7. Run the `./ConfigEngine.sh wp-create-db -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to add a database user registry to the default federated repository. **Note:** Users who are not in an LDAP do
not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

9. Perform the following steps to update the user registry where new users and groups are stored:

   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - `personAccountParent`
      - `groupParent`
      - `personAccountRdnProperties`
      - `groupRdnProperties`

      The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:

      - `personAccountParent=dc=yourco,dc=com`
      - `groupParent=dc=yourco,dc=com`

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `. /ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to delete the old attributes before adding the new attributes.

   E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Run the `. /ConfigEngine.sh wp-query-repository -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to list the names and types of configured repositories.

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See Enabling LDAP security after cluster creation for instructions.

**Parent topic:** Configuring the default federated repository on Linux in a clustered environment

**Related tasks**

Starting and stopping servers, deployment managers, and node agents

Enabling LDAP security after cluster creation
Adding realm support on Linux in a clustered environment

A realm is a group of users from one or more user registries that form a coherent group within IBM® WebSphere® Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

Before configuring realm support, you must add all LDAP user registries and/or database user registries, that you will use to create a single realm or multiple realms, to the federated repository. If you are going to create multiple realms, you must create all required base entries within your LDAP user registries and/or database user registries. All base entry names must be unique within the federated repository. In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add realm support to your user registry model:

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
2. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading:Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - realmName
   - securityUse
   - delimiter
   - addBaseEntry
3. Save your changes to the wkplc.properties file.
4. Run the ./ConfigEngine.sh wp-create-realm -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add a new realm to the Virtual Member Manager configuration. Important: To create multiple realms, ensure that your federated repository contains the required unique base entries. Stop and restart the appropriate servers for your installation environment, and then update the wkplc.properties file with the base entry information and rerun the wp-create-realm task. Repeat these steps until all realms are created.
5. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.
6. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading and then save your changes:Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - realmName
   - realm.personAccountParent
   - realm.groupParent
   - realm.orgContainerParent
7. Run the ./ConfigEngine.sh wp-modify-realm-defaultparents -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to update the default parents per entity type and realm. Important: Stop and restart the appropriate servers for your installation environment before rerunning this task for any additional entity types.
and realms.

8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

9. Optional: Perform the following steps to add additional base entries to the realm configuration; for example, if you had two additional base entries (base entry 1 and base entry 2) to add to the realm you just created, you would update the wkplc.properties file with the information from base entry 1 and then run this task. Then you would update the properties file with the information for base entry 2 and then run this task:
   A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
   B. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading:
      - realmName
      - addBaseEntry
   C. Save your changes to the wkplc.properties file.
   D. Run the ./ConfigEngine.sh wp-add-realm-baseentry -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add additional LDAP base entries to the realm configuration.
   E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID; this step is required if you change the default realm:
   A. Create a new user in the Manage Users and Groups portlet to replace the current WebSphere Application Server administrative user.
   B. Create a new user in the Manage Users and Groups portlet to replace the current WebSphere Portal administrative user.
   C. Create a new group in the Manage Users and Groups portlet to replace the current group.
   D. Run the ./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminId
       -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid task, from the wp_profile_root/ConfigEngine directory, to replace the old WebSphere Application Server administrative user ID and group ID with the new user and group. **Important:** You must provide the full distinguished name (DN) for the newAdminId and newAdminGroupId parameters.
   E. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.
   F. Run the ./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminId
       -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid task to replace the old WebSphere Portal administrative user ID and group ID with the new user and group. **Important:** You must provide the full distinguished name (DN) for the newAdminId and newAdminGroupId parameters.
   G. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.
   H. Optional: Perform the following steps to set the realm you created as the default realm: **Remember:** Only users defined in base entries that exist in the default realm are able to log into WebSphere Portal. If you find that a user cannot log in to WebSphere Portal, check to see if the base entry that contains the user exists in the default realm. You can run the wp-query-realm-baseentry task to see what base entries are part of the default realm. If the default realm is missing
the base entry, run the `wp-add-realm-baseentry` task to add the base entry to the default realm.

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. For `defaultRealmName`, type the `realmName` property value you want to use as the default realm.

C. Save your changes to the `wkplc.properties` file.

D. Run the `.ConfigEngine.sh wp-default-realm -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to set this realm as the default realm.

E. Stop and restart all necessary servers to propagate your changes.

12. Optional: Perform the following steps to query a realm for a list of its base entries:

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. For `realmName`, type the name of the realm you want to query.

C. Save your changes to the `wkplc.properties` file.

D. Run the `.ConfigEngine.sh wp-query-realm-baseentry -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to list the base entries for a specific realm.

13. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. Enter a value for `realmName` or leave blank to update the default realm.

C. Save your changes to the `wkplc.properties` file.

D. Run the `.ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `.ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

E. Stop and restart all necessary servers to propagate your changes.

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring the default federated repository on Linux in a clustered environment

**Related tasks**

- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation
Adapting the attribute configuration

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

After installation, IBM WebSphere Portal has a predefined set of attributes for users and groups. Your LDAP server may have a different set of predefined user and group attributes. To ensure proper communication between WebSphere Portal and your LDAP server, you can configure additional attributes and flag existing attributes as required or unsupported on a per repository basis or for all configure repositories.

LDAP servers can only handle attributes that are explicitly defined in their schema. The LDAP server’s schema is different from the WebSphere Portal schema but the two schemas should match for proper communication between WebSphere Portal and the LDAP server. The task to add the LDAP user registry does some basic attribute configurations depending on the type of LDAP server that you choose. You may, however, still need to adapt the WebSphere Portal configuration to match the LDAP schema; for example, if an attribute is defined in WebSphere Portal but not in the LDAP server, you will need to perform one of the following tasks to resolve this mismatch:

- Flag the attribute as unsupported for the LDAP server
- Introduce an attribute mapping that maps the WebSphere Portal attribute to an attribute defined in the LDAP schema

Perform the following tasks to adapt the attribute configuration to match the configured LDAP server(s) and your business needs:

1. Querying the defined attributes on Linux
   After installing IBM WebSphere Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

2. Adding attributes on Linux in a clustered environment
   The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM WebSphere Portal and your user registry.

3. Mapping attributes on Linux in a clustered environment
   After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

4. Removing attributes
   Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

Parent topic: Choosing your user registry model on Linux in a clustered environment

Related tasks
Adding an LDAP user registry on Linux
Adding an LDAP user registry over SSL on Linux
Configuring a stand-alone LDAP user registry on Linux
Configuring a stand-alone LDAP user registry over SSL on Linux
Querying the defined attributes on Linux

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

**Warning:** If you are using a database user registry or a property extension database, copy the database drivers to the `AppServer_root/lib` directory before executing this script.

Run the `./ConfigEngine.sh wp-query-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, any time during the configuration process or at runtime to query an overview of the currently defined attributes. This task creates the `availableAttributes.html` report, located in the `wp_profile_root/ConfigEngine/log` directory. The report contains one table that lists the available attributes for Users (PersonAccount) and one table that lists the available attributes for Groups. For each configured repository there is a column that indicates if the attribute is flagged as unsupported or if the attribute is mapped to a different LDAP attribute. **Note:** This task does not validate the existence of attributes in the LDAP schema.

**Parent topic:** Adapting the attribute configuration

**Next topic:** Adding attributes on Linux in a clustered environment
Adding attributes on Linux in a clustered environment

The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM® WebSphere® Portal and your user registry.

Perform the following steps to add new attributes to your user registry:

1. Install the required Enterprise Archive (.ear) file on WebSphere Application Server. Select the appropriate task to install the .ear file: Table 1. Steps for installing the .ear file by environment

<table>
<thead>
<tr>
<th>Environment</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone environment</td>
<td>Perform the following steps to install the .ear file in a stand-alone environment: Open a command prompt. Navigate to the wp_profile_root/ConfigEngine directory. Run the ./ConfigEngine.sh wp-la-install-ear -DWasPassword=password task.</td>
</tr>
<tr>
<td>Clustered environment</td>
<td>Perform the following steps to install the .ear file in a clustered environment: Open a command prompt. Navigate to the wp_profile_root/ConfigEngine directory on the primary node. Run the ./ConfigEngine.sh wp-la-install-ear -DWasPassword=dmgr_password -DServerName=dmgr -DNodeName=node_name task. Where node_name is the name of the node where the deployment manager resides; you can find the node_name value in the WebSphere Application Server Administrative Console under System administrator &gt; Deployment Manager &gt; Runtime tab &gt; General Properties &gt; Node Name.</td>
</tr>
</tbody>
</table>

2. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

3. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

4. Enter a value for the following required parameters in the wkplc.properties file under the VMM Property Extension Properties heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - la.providerURL
   - la.propertyName
   - la.entityTypes
   - la.dataType
   - la.multiValued

5. Save your changes to the wkplc.properties file.
6. Run the `./ConfigEngine.sh wp-add-property -DWasPassword=password` task to add the attribute to the user registry.

   **Note:** This task performs an EJB call to WebSphere Application Server, which must authenticate against WebSphere Application Server. Depending on the configuration in the `sas.client.props` file, you may receive a popup window or a command line prompt asking for user identity and password. Enter the WebSphere Application Server user ID and password.

   **Remember:** If you have multiple properties to add, repeat all steps, except for the `wp-la-install-ear` task, until all new attributes are added.

7. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents.*

   If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Adapting the attribute configuration

**Previous topic:** Querying the defined attributes on Linux

**Next topic:** Mapping attributes on Linux in a clustered environment

**Related tasks**

*Starting and stopping servers, deployment managers, and node agents*

*Enabling LDAP security after cluster creation*
Mapping attributes on Linux in a clustered environment

After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs. Perform the following steps to map attributes between WebSphere Portal and your LDAP server; if you have multiple LDAP servers, you will need to perform these steps for each LDAP server:

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

2. Enter a value for one of the following sets of parameters in the wkplc.properties file to identify your LDAP server:
   
   **Note:** Make sure you use the same values you used to configure your LDAP server.

   *Table 1. Identifying your LDAP server in the wkplc.properties file.*

3. Run one of the following tasks to check that all defined attributes are available in the configured LDAP user registry:

   *Table 2. Task to check that all defined attributes are available in the configured LDAP user registry.*

4. Open the ConfigTrace.log file, located in the wp_profile_root/ConfigEngine/log directory, to review the following output for the PersonAccount and Group entity type:
- The following attributes are defined in WebSphere Portal but not in the LDAP server
  - This list contains all attributes that are defined in WebSphere Portal but not available in the LDAP. Flag attributes that you do not plan to use in WebSphere Portal as unsupported. Map the attributes that you plan to use to the attributes that exist in the LDAP; you must also map the uid, cn, firstName, sn, preferredLanguage, and ibm-primaryEmail attributes if they are contained in the list.

- The following attributes are flagged as required in the LDAP server but not in WebSphere Portal
  - This list contains all attributes that are defined as "MUST" in the LDAP server but not as required in WebSphere Portal. You should flag these attributes as required within WebSphere Portal; see the step below about flagging an attribute as either unsupported or required.

- The following attributes have a different type in WebSphere Portal and in the LDAP server
  - This list contains all attributes that WebSphere Portal might ignore because the data type within WebSphere Portal and within the LDAP server do not match.

5. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

6. Enter a value for one of the following sets of parameters in the wkplc.properties file to correct any issues found in the config trace file: Table 3. Parameters to define in the wkplc.properties file to correct any issues found in the config trace file.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>The following parameters are found under the LDAP attribute configuration heading: <strong>Note:</strong> See the properties file for specific information about the required parameters and for advanced parameters.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>standalone.ldap.idstandalone.ldap.attributes.nonSupportedstandalone.ldap.attributes.nonSupported.deletestandalone.ldap.attributes.mapping.ldappNamestandalone.ldap.attributes.mapping.porthalNamestandalone.ldap.attributes.mapping.entityTypes For example, the following values will flag certificate and members as unsupported attributes and will map ibm-primaryEmail to mail and ibm-jobTitle to title for both the PersonAccount and Group entityTypes:</th>
</tr>
</thead>
</table>
7. Save your changes to the \texttt{wkplc.properties} file.

8. Run one of the following tasks to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between WebSphere Portal and the LDAP user registry:

   \textbf{Table 4. Task to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between Portal and the LDAP user registry.}

   \begin{tabular}{|l|l|}
   \hline
   Repository type & Task \tabularnewline
   \hline
   Stand-alone & \texttt{./ConfigEngine.sh wp-update-standalone-ldap-attribute-config -DWasPassword=\text{password} task, from the \text{wp_profile_root}/ConfigEngine directory} \tabularnewline
   \hline
   Federated & \texttt{./ConfigEngine.sh wp-update-federated-ldap-attribute-config -DWasPassword=\text{password} task, from the \text{wp_profile_root}/ConfigEngine directory} \tabularnewline
   \hline
   \end{tabular}

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: \emph{Starting and stopping servers, deployment managers, and node agents}.

10. Optional: Perform the following steps to flag an attribute as either unsupported or required for the entire WebSphere Portal environment instead of just for the specified LDAP:

   A. Enter a value for the following required parameters in the \texttt{wkplc.properties} file: \textbf{Note}: See the properties file for specific information about the required parameters and for advanced parameters.

      - \texttt{user.attributes.required}
      - \texttt{user.attributes.nonsupported}

   B. Save your changes to the \texttt{wkplc.properties} file.

   C. Run the \texttt{./ConfigEngine.sh wp-update-attribute-config -DWasPassword=\text{password} task, from the \text{wp_profile_root}/ConfigEngine directory}.

   D. Stop and restart all necessary servers to propagate your changes.

If you created your clustered environment then performed the steps in this task, you must now run the \texttt{enable-jcr-security} task on the secondary node. See \emph{Enabling LDAP security after cluster creation} for instructions.
Parent topic: Adapting the attribute configuration
Previous topic: Adding attributes on Linux in a clustered environment
Next topic: Removing attributes

Related tasks
Starting and stopping servers, deployment managers, and node agents
Enabling LDAP security after cluster creation
Removing attributes

Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

Perform the following steps to remove an attribute from your database:

**Important:** Do not remove attributes that have already been populated with user values because this can cause database inconsistencies.

**Cluster note:** In a clustered environment, perform the following steps on the deployment manager and then resynch the nodes.

1. Perform the following steps to remove an attribute stored in a property extension database; this is an attribute that was added using the `wp-add-la-property` task.
   
   A. Open the tool you use to edit your database.
   
   B. Verify that your attribute name is available in the LAPROP table.
   
   C. Delete the required attributes from the LAPROP table.
   
   D. Open the `wimxmlextension.xml` file, located in the `wp_profile_root/config/cells/cellname/wim/model` directory.
   
   E. Locate and delete the `propertySchema` definition for the attributes that you deleted from the LAPROP table; for example:

   ```xml
   <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String" multiValued="true" propertyName="attribute_name">
     <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
   </wim:propertySchema>
   ```

   F. Save your changes to the `wimxmlextension.xml` file.

   G. Open the `wimconfig.xml` file, located in the `wp_profile_root/config/cells/cellname/wim/config` directory.

   H. Locate and delete the `propertiesNotSupported` definitions for the attributes that you deleted from the LAPROP table; for example:

   ```xml
   <config:propertiesNotSupported name="attribute_name"/>
   ```

   I. Save your changes to the `wimconfig.xml` file.

   J. Stop and restart the server1 and WebSphere_Portal servers from the `wp_profile_root/bin` directory.

2. Perform the following steps to remove an attribute that is not stored in a property extension database:
   
   A. Open the `wimxmlextension.xml` file.
   
   B. Locate and delete the `propertySchema` definition for the attributes you previously added; for example:

   ```xml
   <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String" multiValued="true" propertyName="attribute_name">
     <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
   </wim:propertySchema>
   ```

   C. Save your changes to the `wimxmlextension.xml` file.

   D. Open the `wimconfig.xml` file.

   E. Locate and delete the stanza that corresponds to the custom attribute you deleted from the `wimextension.xml` file; for example:

   ```xml
   <config:attributes name="attribute_name" propertyName="property_name">
     <config:entityTypes>PersonAccount</config:entityTypes>
   </config:attributes>
   ```

   F. Save your changes to the `wimconfig.xml` file.
G. Stop and restart the server1 and WebSphere_Portal servers.

Parent topic: Adapting the attribute configuration
Previous topic: Mapping attributes on Linux in a clustered environment
Preparing additional nodes on Linux

After installing and configuring your primary node, you can create your secondary nodes. You must install IBM® WebSphere® Portal on each node and then configure the node to access the database and user registry before adding it to the cluster.

- Prerequisites
  - Preparing prerequisite and corequisite software on Linux
  - Preparing your Linux operating system
  - Preparing the primary node on Linux
  - Installing a secondary node after migrating from previous version

Perform the following tasks to prepare your secondary node:

- **Installing WebSphere Portal on Linux on the additional nodes**
  Install IBM WebSphere Portal on your secondary nodes to create a highly available and scalable environment.

- **Choosing the type of additional node to create on Linux**
  If you created a static cluster using the IBM WebSphere Application Server Network Deployment, add additional static nodes to the cluster. If you created a dynamic cluster using IBM WebSphere Virtual Enterprise, add additional dynamic nodes to the cluster.

- **Choosing the type of vertical cluster to create on Linux**
  If you are using the IBM WebSphere Application Server Network Deployment, you can add vertical cluster members to your static cluster. If you are using IBM WebSphere Virtual Enterprise, you can add vertical cluster members to your dynamic cluster.

**Parent topic:** Setting up a cluster on Linux

**Previous topic:** Configuring WebSphere Portal to use a user registry on Linux in a clustered environment

**Next topic:** Tune your servers
Installing WebSphere Portal on Linux on the additional nodes

Install IBM® WebSphere® Portal on your secondary nodes to create a highly available and scalable environment.

Review the Supported hardware and software requirements for your operating system and ensure that all hardware and software matches the minimum product level before installing WebSphere Portal. Review Installation methods, options, and sources before installing to determine your installation source location and installation method. **Note:** If you plan to use IBM WebSphere Application Server Version 7.0, you must purchase and install it before installing WebSphere Portal on the existing WebSphere Application Server Version 7.0 instance.

Perform the following steps to install WebSphere Portal:

1. Type `ping yourserver.yourcompany.com` on a command line to verify that your fully qualified host name is properly configured.
2. If you are installing with an existing WebSphere Application Server instance, ensure it is installed at the supported level and has the following features installed:
   - **Restriction:** Installing WebSphere Portal Version 6.1.0.1 or higher on an existing WebSphere Application Server Version 7.0 instance where WebSphere Portal Version 7.0 was installed is unsupported.
   - **Important:** Besides ensuring that the existing WebSphere Application Server instance is at the supported level, you will also need to ensure that Apache Derby is installed at the supported level before installing WebSphere Portal. See WebSphere Portal detailed system requirements for supported levels.

   - Application Server
   - Administration
     - Scripted Administration
     - Administrative Console
   - Ant and Deployment Tools
     - Deploy Tool
     - Ant Utilities

3. Optional: Perform the following steps if you want to install WebSphere Portal using a non-root user:
   - **Restriction:** Although it is possible to install WebSphere Application Server as a non-root user, there are limitations to this option that can affect WebSphere Portal functionality, which is why you should install WebSphere Application Server as a root user.
   A. Install the current supported version of WebSphere Application Server as a root user.
   B. Open a command prompt and perform the following steps to create a non-root user and to change ownership:

   1. Use the appropriate system commands to create a new group, to create a new user, and to add the new user to the new group.
   2. Run the following tasks to change the rights of the non-root user:
      ```bash
      chmod -R g+rw /opt/IBM
      chgrp -R group_name /opt/IBM
      chmod -R g+wr /tmp
      chgrp -R group_name /tmp
      chmod -R g+wr /var/tmp
      ```
C. Launch the installation program per the next step. During the installation, select the existing WebSphere Application Server using the non-root user and set the installation location to a unique location under the path where non-root permissions were granted.

4. Choose one of the following installation commands based on the installation method you want to use to install WebSphere Portal; see Installation Methods for more information: Advance installation parameters: To customize your installation, you can add various parameters to your installation commands; for example, you can specify your port information. See Advanced installation parameters for information.

Restriction: When defining your installation path, the ONLY supported characters are ASCII letters [A-Z and a-z], numbers [0-9], slashes [/ or \, depending on your operating system], and the dash [-].

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphical user interface</td>
<td>./install.sh</td>
</tr>
<tr>
<td>Console mode</td>
<td>./install.sh -console from the PortalExpress subdirectory</td>
</tr>
<tr>
<td>Silent Install</td>
<td>./install.sh -options &quot;path_to_file/response_filename&quot;, where path_to_file is the full path to the response file and response_filename is the name of the file. A sample install response file (installresponse.txt) and a sample uninstall response file (uninstallresponse.txt) are located in the setup CD root directory. Important: Do not place the response file in a path that contains a space and do not put a space in the file name.</td>
</tr>
</tbody>
</table>

5. Verify your installation was successful; access WebSphere Portal using the http://yourserver:yourport/wps/portal format. Run the .ConfigEngine.sh list-server-ports -DWasPassword=password task from the wp_profile_root/ConfigEngine directory to generate the wp_profile_root/ConfigEngine/log/wp_PortMatrix.txt file that lists WebSphere Portal ports for your installation.

6. Optional: Perform the following steps if you need to change the ports for WebSphere Application Server or WebSphere Portal:

Note: The starting port parameter is required for a successful completion of the modify-ports-by-startport task. Once you specify a start port, this port becomes the base for assigning port values. The code increments this value as each port is assigned, which means that the WebSphere Portal ports will be assigned incrementally starting with the port defined with the modify-ports-by-startport task.

A. Stop the server1 and WebSphere_Portal servers.
B. Run one of the following commands for each server you need to change:

<table>
<thead>
<tr>
<th>Method</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting port number</td>
<td>./ConfigEngine.sh modify-ports-by-startport -DWasPassword=password -DModifyPortsServer=servername -DStartPort=starting port number</td>
</tr>
</tbody>
</table>
C. Restart the server1 and WebSphere_Portal servers.

7. Perform the following steps if you migrated from a previous version of WebSphere Portal:
   A. Log on to the WebSphere Application Server Administrative Console.
   C. Click WP Identification.
   D. Click Custom properties.
   E. If applicable, note the location of the LpidToGupidMapping.properties file.
   F. If the LpidToGupidMapping.properties file exists on the primary node, copy the file to the same location on the secondary node.

**Parent topic:** Preparing additional nodes on Linux

**Related concepts**

Installation methods

IBM Support Assistant Lite for WebSphere Portal

**Related reference**

Advanced installation parameters

---

Port fileNote: Sample port files are available on the Setup disc.

```
./ConfigEngine.sh modify-ports-by-portsfile -DWasPassword=password -DModifyPortsServer=servername -DPortsFile=full path to ports file
```

The following is an example of the information within a port file although the port values will be different based on your environment:

```none
BOOTSTRAP_ADDRESS=10031
SOAP_CONNECTOR_ADDRESS=10033
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=10032
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=10025
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=10036
WC_adminhost=10027
WC_defaulthost=33344
DCS_UNICAST_ADDRESS=10029
WC_adminhost_secure=10039
WC_defaulthost_secure=10035
SIB_ENDPOINT_ADDRESS=10026
SIB_ENDPOINT_SECURE_ADDRESS=10037
SIB_MQ_ENDPOINT_ADDRESS=10030
SIB_MQ_ENDPOINT_SECURE_ADDRESS=10028
ORB_LISTENER_ADDRESS=10034
```
Choosing the type of additional node to create on Linux

If you created a static cluster using the IBM® WebSphere® Application Server Network Deployment, add additional static nodes to the cluster. If you created a dynamic cluster using IBM WebSphere Virtual Enterprise, add additional dynamic nodes to the cluster.

Choose one of the following options to add additional nodes to your cluster:

- **Adding additional nodes to the static cluster on Linux**
  After installing IBM WebSphere Portal on your secondary node, you must add the node to the cluster to create a highly available environment.

- **Adding an additional node to an existing dynamic cluster**
  After creating your dynamic cluster, you can add additional nodes to expand the capacity of the dynamic cluster. Install and configure the additional node, then add it to the existing dynamic cluster node group, and then add it to the existing IBM WebSphere Virtual Enterprise dynamic cluster.

**Parent topic:** Preparing additional nodes on Linux
Adding additional nodes to the static cluster on Linux

After installing IBM® WebSphere® Portal on your secondary node, you must add the node to the cluster to create a highly available environment.

Perform the following steps to add the secondary node to the cluster:

1. Perform the following steps on the secondary node to access your database server:
   - Note: All properties files are located in the `wp_profile_root/ConfigEngine/properties` directory.
   - For JDBC Type 2 connections only, ensure that the database client software is installed on the secondary node using the same settings as the primary node and that you can connect to the remote database. It is not necessary to create databases or users when configuring secondary nodes.
   - **DB2 users:** When using DB2, you must catalog the TCP/IP node and all databases used on each secondary node. For more information refer to Preparing DB2.
   - For JDBC Type 4 connections only, copy the JDBC Type 4 jar files to the secondary node from the DB2 server.
   - Set all database properties in the `wkplc_dbtype.properties` and `wkplc_comp.properties` files with the appropriate values from the primary node.

2. Run the following tasks to validate your database settings:
   - A. `./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password`
   - B. `./ConfigEngine.sh validate-database-connection -DWasPassword=password -DTransferDomainList=release,customization,community,jcr,feedback,likeminds`

3. Stop the server1 and WebSphere_Portal servers on the secondary node and ensure that the following parameters are set correctly in the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory of the secondary node:
   - **Note:** Although you can add these parameters directly to any task that you run while creating your cluster, you may want to add them to the properties file while creating your cluster and then remove them when you are finished to keep your environment secure.
   - A. Verify that `WasUserid` is set to your deployment manager administrator ID.
   - B. Verify that `WasPassword` is set to your deployment manager administrator password.
   - C. Verify that `PortalAdminId` is set to your WebSphere Portal administrator ID.
   - D. Verify that `PortalAdminPwd` is set to your WebSphere Portal password.
   - E. Verify that `WasRemoteHostName` is set to the fully qualified host name of the deployment manager.
   - F. Verify that `WasSoapPort` is set to the SOAP port that the deployment manager is using; the default value is 8879.
   - G. Verify that `PrimaryNode` is set to `false`.
   - H. Verify that `ClusterName` is set to the primary node’s `ClusterName`.

4. Run the `./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=dmgr_password` task, from the `wp_profile_root/ConfigEngine` directory of the secondary node.

5. Optional: If you configured a database user registry or a property extension (lookaside) database, run the following task to set up access to the database drivers:
   - **Note:** You will also need to run this task if you migrated the primary node from a release prior to Version 6.1.0, even if you did not configure a database user registry or a property extension (lookaside) database.
   - A. Set the property value for `federated.db.DbType` if using a database user registry or set the property value for `la.DbType` if using a property extension database in the `wkplc.properties` file.
   - **Note:** If you migrated the primary node from a version prior to Version 6.1.0 and you are not using a database user registry or property extension database, set the property value for `federated.db.DbType` or `la.DbType` as appropriate.
registry or a property extension database, set the property value for `federated.dbDbType` to the same value that is in the wkplc.properties file on the primary node.

6. Run the `./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=dmgr_password -DDbDomain=la|federated.db -DVmmNodeName=node_name -Db_type.NodeDbLibrary=local full path of the database jars` task from the wp_profile_root/ConfigEngine directory to create the variable used to access the VMM database jars.

   **Note:** `VmmNodeName` is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The `db_type` in `db_type.NodeDbLibrary` should be set to the type of database you are using, for example `db2`.

7. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell's user registry. Run the `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=dmgr_password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task, from the wp_profile_root/ConfigEngine directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

   **Attention:** The password value for `-DWasPassword` is the Deployment Manager administrative password.

   **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

   **Fast path:** If the value for `newAdminGroupId` contains a space; for example `Software Group`, open the wkplc.properties file and add the values for `newAdminId`, `newAdminPw`, and `newAdminGroupId`. Save your changes and run the ConfigEngine.bat wp-change-portal-admin-user -DWasPassword=dmgr_password task.

   **Important:** If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the wp-change-portal-admin-user task until after the cluster-node-config-cluster-setup (static cluster) or cluster-node-config-dynamic-cluster-setup (dynamic cluster) task completes. After running the wp-change-portal-admin-user task, start or restart the WebSphere_Portal server to use the updated administrator user ID.

   **Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the `wp-change-portal-admin-user` task. `-DnewAdminPw` is an optional parameter to update the Administrative password in the wkplc.properties file if required.

8. Optional: Open the wkplc.properties file and change the `ServerName` parameter from the default WebSphere_Portal_nodename value to a value that meets your business needs. Do not change the parameter to WebSphere_Portal as this is the primary node value.

9. Run the `./ConfigEngine.sh cluster-node-config-cluster-setup -DWasPassword=dmgr_password` task to add the node to your cluster.

10. Wait at least 30 minutes before stopping and restarting the deployment manager because the node agent runs a background thread to expand all EAR files into their target directories. **Note:** Attempting to restart the deployment manager while the expansion is in process can result in an unusable application. Although the time to complete an EAR expansion is environment-dependent, a 30 minute wait is sufficient for a typical environment.

11. Perform the following steps to access the Web Content Management content through an external Web server:

    A. Log on to the deployment manager administrative console.

    B. Select **Environment > WebSphere Variables**.

    C. From the **Scope** drop-down menu, select the `Node=node_name, Server=servername` option to narrow the scope of the listed variables, where `Node=node_name` is the node that contains the application server.
D. Update the **WCM_HOST** variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.

E. Update the **WCM_PORT** variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.

F. Perform the following steps to synchronize the node with the deployment manager:
   1. Select **System Administration > Nodes.**
   2. Select the node that you want to synchronize from the list.
   3. Click **Full Resynchronize.**

G. Log off of the deployment manager administrative console.

12. If you entered passwords in any of the properties files while creating your cluster, you should remove them for security purposes. See "Deleting passwords from properties files" under Related tasks for information.

**Parent topic:** Choosing the type of additional node to create on Linux

**Related tasks**

Deleting passwords from properties files
Adding an additional node to an existing dynamic cluster

After creating your dynamic cluster, you can add additional nodes to expand the capacity of the dynamic cluster. Install and configure the additional node, then add it to the existing dynamic cluster node group, and then add it to the existing IBM® WebSphere® Virtual Enterprise dynamic cluster.

On all WebSphere Portal nodes, install WebSphere Virtual Enterprise and augment the wp_profile profile. See Installing and configuring the product for information about installing WebSphere Virtual Enterprise. See Augmenting profiles for information about augmenting the Deployment Manager and wp_profile profiles.

- Prerequisites
  - PK97393 for all WebSphere Application Server Versions prior to 6.1.0.5
  - Installing and configuring the product:
  - Augmenting profiles

Perform the following steps to add an additional node to an existing WebSphere Virtual Enterprise dynamic cluster:

1. Perform the following steps on the secondary node to access your database server: **Note:** All properties files are located in the wp_profile_root/ConfigEngine/properties directory.
   A. For JDBC Type 2 connections only, ensure that the database client software is installed on the secondary node using the same settings as the primary node and that you can connect to the remote database. It is not necessary to create databases or users when configuring secondary nodes. **DB2 users:** When using DB2, you must catalog the TCP/IP node and all databases used on each secondary node. For more information refer to Preparing DB2.
   B. For JDBC Type 4 connections only, copy the JDBC Type 4 jar files to the secondary node from the DB2 server.
   C. Set all database properties in the wkplc_dbtype.properties and wkplc_comp.properties files with the appropriate values from the primary node.

2. Run the following tasks to validate your database settings:
   A. ```/ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=```
   B. ```/ConfigEngine.sh validate-database-connection -DWasPassword= -DTransferDomainList=release,customization,community,jcr,feedback,likeminds```

3. Stop the server1 and WebSphere_Portal servers on the secondary node and ensure that the following parameters are set correctly in the wkplc.properties file, located in the wp_profile_root\ConfigEngine\properties directory of the secondary node: **Note:** Although you can add these parameters directly to any task that you run while creating your cluster, you may want to add them to the properties file while creating your cluster and then remove them when you are finished to keep your environment secure.
   A. Verify that WasUserid is set to your deployment manager administrator ID.
   B. Verify that WasPassword is set to your deployment manager administrator password.
   C. Verify that PortalAdminId is set to your WebSphere Portal administrator ID.
   D. Verify that PortalAdminPwd is set to your WebSphere Portal password.
   E. Verify that WasRemoteHostName is set to the fully qualified host name of the deployment manager.
   F. Verify that WasSoapPort is set to the SOAP port that the deployment manager is using; the default value is 8879.
   G. Verify that PrimaryNode is set to false.
H. Verify that **ClusterName** is set to the primary node’s **ClusterName**.

4. Run the `./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=dmgr_password` task, from the `wp_profile_root/ConfigEngine` directory of the secondary node.

5. Optional: If you configured a database user registry or a property extension (lookaside) database, run the following task to set up access to the database drivers:

   **Note:** You will also need to run this task if you migrated the primary node from a release prior to Version 6.1.0, even if you did not configure a database user registry or a property extension (lookaside) database.

   A. Set the property value for **federated.db.DbType** if using a database user registry or set the property value for **la.DbType** if using a property extension database in the `wkplc.properties` file.

      **Note:** If you migrated the primary node from a version prior to Version 6.1.0 and you are not using a database user registry or a property extension database, set the property value for **federated.db.DbType** to the same value that is in the `wkplc.properties` file on the primary node.

   B. Run the `./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=dmgr_password -DDbDomain=la|federated.db -DVmmNodeName=node_name -Db_type.NodeDbLibrary=local full path of the database jars` task from the `wp_profile_root/ConfigEngine` directory to create the variable used to access the VMM database jars.

      **Note:** **VmmNodeName** is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The **db_type** in **db_type.NodeDbLibrary** should be set to the type of database you are using, for example **db2**.

6. Run the `./ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=dmgr_password` task. **Note:** If you want to customize the WebSphere Portal server name (WebSphere_Portal) for your additional node, set the **ServerName** parameter in the `wkplc.properties` file after running the `cluster-node-config-post-federation` task.

7. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell’s user registry. Run the `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task, from the `wp_profile_root/ConfigEngine` directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

   **Attention:** The password value for `-DWasPassword` is the Deployment Manager administrative password.

   **Important:** You must provide the full distinguished name (DN) for the **newAdminId** and **newAdminGroupId** parameters.

   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

   **Fast path:** If the value for **newAdminGroupId** contains a space; for example **Software Group**, open the `wkplc.properties` file and add the values for **newAdminId**, **newAdminPw**, and **newAdminGroupId**. Save your changes and run the `ConfigEngine.bat wp-change-portal-admin-user -DWasPassword=dmgr_password` task.

   **Important:** If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user` task until after the `cluster-node-config-cluster-setup` (static cluster) or `cluster-node-config-dynamic-cluster-setup` (dynamic cluster) task completes. After running the `wp-change-portal-admin-user` task, start or restart the WebSphere_Portal server to use the updated administrator user ID.

   **Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the `wp-change-portal-admin-user` task. `-DnewAdminPw` is an optional parameter to update the Administrative password in the `wkplc.properties` file if required.

8. Log on to the deployment manager administrative console.

9. Perform the following steps to add members to the node group:

   A. Click **System administration > Node groups**.
B. Click on the name of the node group that you want to add members to.
C. Click **Node group members** under Additional Properties.
D. Click **Add**.
E. Select the additional node you want to add into the node group and then click **Add**.
F. Click the **Save** link to save your changes to the master configuration.

10. Wait at least 30 minutes before stopping and restarting the deployment manager because the node agent runs a background thread to expand all EAR files into their target directories. **Note:** Attempting to restart the deployment manager while the expansion is in process can result in an unusable application. Although the time to complete an EAR expansion is environment-dependent, a 30 minute wait is sufficient for a typical environment.

11. Define or verify the following parameters in the `wkplc.properties` file, located in the `wp_profile_root` directory:
   A. Set **ClusterName** to the name of the existing dynamic cluster.
   B. Verify that **CellName** is set to the deployment manager cell.
   C. Verify that **NodeName** is set to the local WebSphere Portal node.
   D. Set **ServerName** to the server that will be used for the dynamic cluster member on this node. **Note:** Log on to the deployment manager administrative console and click **Dynamic Clusters > PortalCluster > Dynamic cluster members** to find the name of the server used for the dynamic cluster.
   E. Verify that **PrimaryNode** is set to false because this is an additional node.

12. Run the `./ConfigEngine.sh cluster-node-config-dynamic-cluster-setup -DWasPassword=dmgr_password` task to add the new member to the existing dynamic cluster. **Note:** This task may display a warning message since the cluster already exists but it will proceed to create the new dynamic cluster; therefore, the warning can be ignored.

13. Perform the following steps to start the cluster member:
   A. Log on to the Deployment Manager administrative console.
   B. Navigate to **Servers > Dynamic clusters > cluster name > Dynamic cluster members**.
   C. Select the cluster member and click **Start**.
   D. Log off of the deployment manager administrative console.

14. Perform the following steps to access the Web Content Management content through an external Web server:
   A. Log on to the deployment manager administrative console.
   B. Select **Environment > WebSphere Variables**.
   C. From the **Scope** drop-down menu, select the **Node=nodename, Server=servername** option to narrow the scope of the listed variables, where **Node=nodename** is the node that contains the application server.
   D. Update the **WCM_HOST** variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.
   E. Update the **WCM_PORT** variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.
   F. Perform the following steps to synchronize the node with the deployment manager:
      1. Select **System Administration > Nodes**.
      2. Select the node that you want to synchronize from the list.
      3. Click **Full Resynchronize**.
   G. Log off of the deployment manager administrative console.

15. Run the following tasks to propagate your changes: **Note:** `WebSphere_Portal_nodename` is the name of the node’s WebSphere Portal server; but if you customized the server name, the default name changes. If you are uncertain of the server name, run the `serverstatus -all` task to get a list of the server names and their status.
   A. `./stopManager.sh -username admin_userid -password admin_password`, from the `dmgr_profile_root\bin` directory
   B. `./stopNode.sh -username admin_userid -password admin_password`, from the `wp_profile_root\bin` directory
   C. `./stopServer.sh WebSphere_Portal_nodename -username admin_userid -password admin_password`
D. ./startManager.sh, from the dmgr_profile_root\bin directory
E. ./startNode.sh, from the wp_profile_root/bin directory
F. ./startServer.sh WebSphere_Portal_nodename

16. If you entered passwords in any of the properties files while creating your cluster, you should remove them for security purposes. See "Deleting passwords from properties files" under Related tasks for information.

Parent topic: Choosing the type of additional node to create on Linux

Related tasks
Deleting passwords from properties files
Choosing the type of vertical cluster to create on Linux

If you are using the IBM® WebSphere® Application Server Network Deployment, you can add vertical cluster members to your static cluster. If you are using IBM WebSphere Virtual Enterprise, you can add vertical cluster members to your dynamic cluster.

Choose one of the following options to add vertical cluster members to your cluster.

- **Adding vertical cluster members to a static cluster on Linux**
  
  You can add vertical cluster members to share the workload demands of your cluster across multiple members running on the same physical machine.

- **Adding vertical cluster members to a dynamic cluster on Linux**
  
  You can add vertical cluster members to share the workload demands of your cluster across multiple members running on the same physical machine.

Parent topic: Preparing additional nodes on Linux
Adding vertical cluster members to a static cluster on Linux

You can add vertical cluster members to share the workload demands of your cluster across multiple members running on the same physical machine.

Perform the following steps to add a vertical cluster member to your clustered environment:

1. Log into the deployment manager administrative console.
2. Depending on your version of WebSphere Application Server, make the appropriate selection from the navigation:
   - For WebSphere Application Server Version 6.1: Click Servers > Clusters in the console navigation tree, select the cluster name, and then click Cluster members from the list of additional properties.
   - For WebSphere Application Server Version 7.0: Click Servers > Clusters > WebSphere application server clusters > cluster_name > Cluster members.
   - Click New to create the cluster member.
     - Define the name of cluster member. Note: Do not use spaces in the cluster member name.
     - Select an existing node where IBM® WebSphere® Portal is installed.
     - Check the box Generate Unique HTTP Ports.
     - Click Add Member and then click Next to view the summary.
     - Important: You must update the virtual host entries for the new port created when adding a cluster member. You can do this by updating the default_host virtual host in the administrative console and adding a new alias entry for the port number (use an "*" wildcard character for the host name). See Configuring virtual hosts for information.
     - Click Finish, and save the changes.
3. The new cluster topology can be viewed from the Servers > Cluster Topology view. If using WebSphere Application Server Version 7.0, the new cluster topology can be viewed from the Servers > Clusters > Cluster Topology view.
4. Perform the following steps to enable cache replication:
   - From the deployment manager Administrative Console, navigate to Servers > Application servers and then click the new vertical cluster member(s). If using WebSphere Application Server Version 7.0, navigate to Servers > Server Types > WebSphere application servers and then click the new vertical cluster member(s).
   - Click Dynamic cache service under Container services.
   - Change Cache size to 3000 entries.
   - Check the Enable cache replication check box.
   - Select NOT_SHARED from the Replication type drop-down menu.
   - Click OK.
   - Click Save to save your changes to the master configuration.
5. Perform the following steps on each vertical cluster member to clean up the server-scoped resources, caches, and resource providers:
   - Run the ./ConfigEngine.sh cluster-node-config-vertical-cluster-setup -DServerName=unique vertical cluster servername -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, where unique vertical cluster servername is the name you specified when you created the cluster member.
   - Restart the vertical cluster member referenced in the cluster-node-config-vertical-cluster-setup task.
7. Perform the following steps to access the Web Content Management content through an external Web server:
   A. Log on to the deployment manager administrative console.
   B. Select Environment > WebSphere Variables.
   C. From the Scope drop-down menu, select the Node=nodename, Server=servername option to narrow the scope of the listed variables, where nodename is the node that contains the application server.
   D. Update the WCM_HOST variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.
   E. Update the WCM_PORT variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.
   F. Perform the following steps to synchronize the node with the deployment manager:
      1. Select System Administration > Nodes.
      2. Select the node that you want to synchronize from the list.
      3. Click Full Resynchronize.
   G. Log off of the deployment manager administrative console.

8. Save your changes and resynchronize the nodes.
   A. In the administrative console for the deployment manager, click Save on the task bar, and save your administrative configuration.
   B. Select System Administration > Nodes, select the node from the list, and click Full Resynchronize.

9. Regenerate the Web server plug-in.
   A. Regenerate the Web server plug-in using the deployment manager administrative console.
   B. If you are using a remote Web server, copy the updated plug-in configuration file (plugin-cfg.xml) to the Web server's plug-in configuration directory.

10. Stop and start the Web server.

Parent topic: Choosing the type of vertical cluster to create on Linux
Adding vertical cluster members to a dynamic cluster on Linux

You can add vertical cluster members to share the workload demands of your cluster across multiple members running on the same physical machine.

Perform the following steps to add a vertical cluster member to your clustered environment:

1. Open a browser and enter `http://DM01:9060/ibm/console` in the address bar to access the administrative console on the deployment manager, where `DM01` is the deployment manager node or host name. The port number might differ based on your installation.

2. Perform the following steps to allow vertical clusters on your dynamic cluster:
   A. Navigate to `Servers > Dynamic clusters` and select the appropriate dynamic cluster.
   B. Select the **Allow more than one instance to start on the same node** check box under **Vertical stacking of instances on node**.
   C. Enter a new value in the **Number of instances** text box to determine the number of vertical cluster members allowed on each node.
   D. Click **Apply** and then click **Save** to save the changes to the master configuration.

3. Perform the following steps to enable cache replication:
   A. From the deployment manager Administrative Console, navigate to `Servers > Application servers` and then click the new vertical cluster member(s). If using WebSphere Application Server Version 7.0, navigate to `Servers > Server Types > WebSphere application servers` and then click the new vertical cluster member(s).
   B. Click **Dynamic cache service** under **Container services**.
   C. Change **Cache size** to 3000 entries.
   D. Check the **Enable cache replication** check box.
   E. Select **NOT_SHARED** from the **Replication type** drop-down menu.
   F. Click **OK**.
   G. Click **Save** to save your changes to the master configuration.

4. Perform the following steps on each vertical cluster member to clean up the server-scoped resources, caches, and resource providers:
   A. Run the `./ConfigEngine.sh cluster-node-config-vertical-cluster-setup -DServerName=unique vertical cluster servername -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, where `unique vertical cluster servername` is the name you specified when you created the cluster member.
   B. Restart the vertical cluster member referenced in the **cluster-node-config-vertical-cluster-setup** task.

5. Perform the following steps to access the Web Content Management content through an external Web server:
   A. Log on to the deployment manager administrative console.
B. Select **Environment > WebSphere Variables**.

C. From the **Scope** drop-down menu, select the `Node=nodename, Server=servername` option to narrow the scope of the listed variables, where `nodename` is the node that contains the application server.

D. Update the **WCM_HOST** variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.

E. Update the **WCM_PORT** variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.

F. Perform the following steps to synchronize the node with the deployment manager:
   1. Select **System Administration > Nodes**.
   2. Select the node that you want to synchronize from the list.
   3. Click **Full Resynchronize**.

G. Log off of the deployment manager administrative console.

6. Save your changes and resynchronize the nodes.
   A. In the administrative console for the deployment manager, click **Save** on the task bar, and save your administrative configuration.
   B. Select **System Administration > Nodes**, select the node from the list, and click **Full Resynchronize**.

7. Stop and start the Web server.

**Parent topic:** Choosing the type of vertical cluster to create on Linux
Tune your servers

Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

- Prerequisites
  - Preparing prerequisite and corequisite software on Linux
  - Preparing your Linux operating system
  - Preparing the primary node on Linux
  - Base Portal Tuning Scenarios

Tuning a WebSphere Portal environment involves tuning and configuring the various systems and components of the environment. The tuning guide provides general concepts and details specifics configuration instructions. Instructions are included for:

- Configuring the application server and the resources defined for that application server
- Determining the cloning strategy for expanding or extending the environment
- Tuning the database(s) and database server
- Tuning the directory server and its database
- Tuning the Web server
- Tuning the operating system and network
- Tuning the WebSphere Portal services

Parent topic: Setting up a cluster on Linux
Previous topic: Preparing additional nodes on Linux
Next topic: Configuring search in a cluster on Linux

Related information
- WebSphere Portal and Lotus Web Content Management 6.1.x Performance Tuning Guide
- Tuning a Cluster Environment
- IBM WebSphere Portal Performance Troubleshooting Guide
Configuring search in a cluster on Linux

IBM® WebSphere® Portal provides two distinct search capabilities. You can utilize both types of search capabilities in a clustered environment.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on Linux
  - Preparing your Linux operating system
  - Preparing the primary node on Linux
  - Tune your servers

- **Configuring Portal Search in a cluster on Linux**
  To support Portal Search in a clustered environment, you must install and configure search for remote search service on an IBM WebSphere Application Server node that is not part of the IBM WebSphere Portal cluster.

- **Configuring JCR search in a cluster on Linux**
  To enable search in a cluster for content stored in the JCR database, you must configure each server in the cluster to access a shared directory. JCR-based content includes content created with Web Content Management or Personalization.

Parent topic: Setting up a cluster on Linux

Previous topic: Tune your servers

Next topic: Setting up multiple clusters on Linux
Configuring Portal Search in a cluster on Linux

To support Portal Search in a clustered environment, you must install and configure search for remote search service on an IBM® WebSphere® Application Server node that is not part of the IBM WebSphere Portal cluster.

To install and configure the search service remotely, perform the following tasks:

1. Install and configure the search service to work remotely, that is, on a remote WebSphere Application Server node which is not part of the portal cluster. You can provide the remote search service either as an EJB or as a Web service via SOAP. Deploy the appropriate EJB or SOAP EAR file on the remote WebSphere Application Server node. For details about how to do this, refer to the WebSphere Application Server documentation.
2. Configure the search portlets for remote search service so that they access the remote server accordingly.

Notes:
1. If you have configured a remote search service for a portal cluster, you need to configure the default location for search collections to a directory on the remote server that has write access.
2. The portal site default search collection is created only once at the first time when an administrator selects the search administration portlet Manage Search. If this occurred before you configure the portlet for remote search, then the default portal site search collection is only available on the primary node of the cluster, but not on the remote server. In this case you need to recreate the portal site collection to make it available for search on all nodes of the cluster.

Parent topic: Configuring search in a cluster on Linux

Related concepts
Planning and preparing for Portal Search
Using remote search service

Related tasks
Configuring Portal Search for remote search service
Configuring the Search and Browse portlet for remote search service
Configuring the default location for search collections
Creating or resetting the portal site collection
Configuring the primary node to communicate with the deployment manager on Linux
Configuring a remote search service
Configuring JCR search in a cluster on Linux

To enable search in a cluster for content stored in the JCR database, you must configure each server in the cluster to access a shared directory. JCR-based content includes content created with Web Content Management or Personalization. Create a shared directory called `jcr/search` on a server in the network and ensure that each node in the cluster and the Deployment Manager has network access to the directory.

**Note:** If you are creating content in a clustered environment using the authoring portlet provided with Web Content Management, additional configuration steps are required to enable content created by these content features to be searchable in a cluster.

Perform the following steps on each server in the cluster to configure Search in a clustered environment:

1. Edit the `icm.properties` file, located in the `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm` directory.
2. Change the value of the `jcr.textsearch.indexdirectory` property to the shared directory; for example, `jcr.textsearch.indexdirectory=\\your_server\your_share\jcr\search`. You can specify the shared directory value in one of the following formats: **Table 1. The format for the shared directory.**

<table>
<thead>
<tr>
<th>Format</th>
<th>Shared directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal Naming Convention (UNC)</td>
<td><code>\\\your_server\your_share\jcr\search</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>\\\hostname.example.com\share\jcr\search</code></td>
</tr>
<tr>
<td>Mounted resource format (with forward slashes)</td>
<td><code>/your_share/jcr/search</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>/mnt/jcr/search</code></td>
</tr>
<tr>
<td><strong>Important:</strong> This format requires that you mount the shared directory to the local server (for example, through a mapped network drive or a mounted directory). When using the mounted resource format, always use forward slashes instead of back slashes, regardless of the native operating system path format.</td>
<td></td>
</tr>
</tbody>
</table>

3. Required: Perform the following steps to delete the default search collections from the Manage Search portlet:

   A. Log on to WebSphere Portal as an administrator.
   B. Click **Administration > Search Administration > Manage Search**.
   C. Click **Search Collections**.
   D. Click the **Delete Collection** icon for the **Portal Content** search collection.
   E. Click **OK**.
   F. Restart the WebSphere_Portal server.
   G. Go to the Manage Search portlet and confirm that the **Portal Content** search collection was deleted.

**Parent topic:** Configuring search in a cluster on Linux
Setting up multiple clusters on Linux

The majority of the steps to build another cluster are the same as when building the first cluster in the same cell, with a few exceptions. Basically, the new profile will be designated as the primary profile, using IBM® WebSphere® Portal clustering terminology, and will then be used as the basis for the new cluster definition. This duplicates the build process of the first cluster in the cell. During the federation process, if any applications on this new node already exist in the cell (because they are in use by the first cluster), then Deployment Manager will not allow them to be added. After federation, the applications that already exist in the cell are not mapped to the WebSphere_Portal server on the newly federated node, and thus the existing applications must be remapped to this newly federated server to restore its application list. Therefore, depending on the configuration of the new profile, there will likely be some combination of applications shared with the other existing cluster, and some applications unique to this new profile.

Before attempting alternative approaches for building multiple portal-based clusters within a single cell, please contact IBM.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on Linux
  - Preparing your Linux operating system
  - Preparing the primary node on Linux
  - Tune your servers

- **Installing multiple clusters in a single cell on Linux**
  Create a new, independent IBM WebSphere Portal cluster in a cell where a WebSphere Portal cluster already exists.

- **Routing requests across clusters on Linux**
  The HTTP Server plug-in that comes with IBM WebSphere Application Server is typically used to balance requests for applications across members of the cluster. While an application can be mapped to more than one cluster, automatic plug-in generation does not provide routing or balancing traffic for the same application across multiple clusters. Multiple cluster environments with shared applications therefore cannot rely solely on WebSphere Application Server automatic plug-in generation to be able to route requests using a web server. One option in this scenario is developing a customized method for defining and maintaining the plug-in configuration file used by the Web server to provide for the required routing of user requests.

**Parent topic:** Setting up a cluster on Linux

**Previous topic:** Configuring search in a cluster on Linux

**Next topic:** Sharing database domains between clusters on Linux
Installing multiple clusters in a single cell on Linux

Create a new, independent IBM® WebSphere® Portal cluster in a cell where a WebSphere Portal cluster already exists. In the following steps, Cluster A will be used to describe the existing cluster. Portal B will be used to describe the new server profile that will be the basis for the new cluster definition, Cluster B. Perform the following steps to install multiple clusters in a single cell:

1. Upgrade Cluster A, including the Deployment Manager node, to the current, supported hardware and software levels and to the current version of IBM WebSphere Portal.
2. Install and configure Portal B; see the "Preparing the primary node" topic for the appropriate operating system for details. Important: Maintain the same number of data sources with identical names to the Cluster A data sources so that data source bindings in the applications can be resolved on every cluster in which they run. If implementing database sharing across the clusters, the above statement refers to both the shared and non-shared domains; all domains should use the same names.
3. Optional: Using the same database user ID and password for each identically named domain/data source will allow the existing JAAS Authentication Aliases to be functional. If unique database user ID and password are required, additional manual configuration is needed to create new JAAS Authentication Aliases for each data source and map these accordingly. On the primary node of Cluster A, run the ./ConfigEngine.sh create-alias-multiple-cluster -DauthDomainList=release,jcr -DWasPassword=dmgr_password task from the wp_profile_root/ConfigEngine directory to create the new JAAS Authentication Aliases. Where authDomainList is set to a list of domains which use unique database user ID and passwords and those domain properties are set correctly in the wkplc_comp.properties file, including user ID and password.
4. Optional: If necessary, upgrade Portal B to the current cumulative fix.
5. Run the ./ConfigEngine.sh mapped-app-list-create -DWasPassword=password /ConfigEngine directory to build an inventory list of Portal B enterprise applications and portlets.
6. Stop the server1 and WebSphere_Portal servers on Portal B and ensure that the following parameters are set correctly in the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory:
   - Set WasSoapPort to the deployment manager's port.
   - Set WasRemoteHostName to the full host name of the deployment manager.
7. If the Deployment Manager is configured to use a stand-alone LDAP user registry, update the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory, on the primary node with the stand-alone LDAP user registry property values from the Deployment Manager. You can find these settings under the VMM Stand-alone LDAP configuration heading. Important: Ensure that you set WasUserId and WasPassword to the Deployment Manager user ID and password.
8. Run the ./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=password task, from the wp_profile_root/ConfigEngine of the primary node.
9. Run the ./ConfigEngine.sh map-apps-to-server -DWasPassword=password task to determine which applications from the inventory list are no longer mapped to Portal B. The task uses the application profiles already in the cell to restore the mappings. Wait 30 minutes after running this task to allow all EAR files to expand before proceeding to the next step.
10. Perform the following steps to federate Portal B into the deployment manager cell:
   A. Ensure that all database parameters are correctly set, including passwords, in the wkplc_comp.properties and wkplc_dbtype.properties files.
B. Run the \\ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=dmgr_password\ task.

C. After running the cluster-node-config-post-federation\ task, wait at least 30 minutes to allow all EAR files to expand.

D. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell’s user registry. Run the \\ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=\password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid\ task, from the wp_profile_root/ConfigEngine directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

Additional parameter for stopped servers: This task verifies the user against a running server instance. If the server is stopped, add the -Dskip.ldap.validation=true parameter to the task to skip the validation.

Important: If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the wp-change-portal-admin-user\ task until after the cluster-node-config-cluster-setup (static cluster) or cluster-node-config-dynamic-cluster-setup (dynamic cluster) task completes. After running the wp-change-portal-admin-user\ task, start or restart the WebSphere Portal server to use the updated administrator user ID.

Note: The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the wp-change-portal-admin-user\ task. -DnewAdminPw is an optional parameter to update the Administrative password in the wkplc.properties file if required.

E. From the administrative console, click System Administration > Node Agents.

F. Check the box next to the required node agent and then click Restart.

G. Stop and restart the deployment manager.

H. Stop and restart the WebSphere Portal server on Portal B

11. Restart the WebSphere Portal server on Cluster A. Verify that Cluster A is functionally intact by spot checking pages and portlets and then verify that Portal B is functionally intact by spot checking pages and portlets that you deployed into Portal B before it was federated. Any discrepancies or errors should be corrected now before continuing. Note: If Portal B is using a non-default Portal server administrative ID, not wpsadmin, the server will not be functional until the cluster configuration is complete and the Portal administrative ID has been configured to match the Cells security settings.

12. Choose one of the following options to define a cluster using Portal B as the basis:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>

1778
Perform the following steps to define a static cluster using **Portal B** as the basis: Run the `./ConfigEngine.sh cluster-node-config-cluster-setup -DWasPassword=dmgr_password` task. Configure the cluster to use an external Web server to take advantage of features such as workload management. Choose one of the following options:

- Configuring a Web server and an application server on separate machines (remote)
- Configuring a Web server and an application server profile on the same machine
- Configuring a Web server and a deployment manager profile on the same machine

**Note:** Start with the step about launching the Plug-ins installation wizard. Perform the following steps to access the Web Content Management content through an external Web server: Log on to the deployment manager administrative console. Select **Environment > WebSphere Variables**. From the **Scope** drop-down menu, select the `Node=nodename, Server=servername` option to narrow the scope of the listed variables, where `Node=nodename` is the node that contains the application server. Update the **WCM_HOST** variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router. Update the **WCM_PORT** variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router. Save your changes and then restart the deployment manager, the node agent(s), server1, and the WebSphere_Portal servers. Change the task list settings to point to the cluster instead of a single server if using Portal business process integration. See one of the following sections in the Information Center for information:

- Cross cell: Adding a BPI-enabled portal server to a managed cell in a cross-cell setup
- Single cell: Adding a BPI-enabled portal server to a managed cell in a single-cell setup
Dynamic cluster

Perform the following steps to define a dynamic cluster using **Portal B** as the basis:

1. Log on to the deployment manager administrative console.
2. Perform the following steps to create a node group:
   - Click **New**. Type the node group **Name**. Type any information about the node group in the **Description** text box. Click **OK**. Click the **Save** link to save your changes to the master configuration.
3. Perform the following steps to add members to the node group:
   - Click **System administration > Node groups**. Click on the name of the node group that you want to add members to. Click **Node group members** under **Additional Properties**. Click **Add**. Select the primary node and then click **Add**. Click the **Save** link to save your changes to the master configuration.
4. Perform the following steps to create a dynamic cluster in the node group:
   - Click **Servers > Dynamic clusters**. Click **New**. Select WebSphere Application Server from the **Server Type** pull-down menu and then click **Next**. Type the cluster name in the **Dynamic cluster name** text box and then click **Next**. Type the same value that you provided for the **ClusterName** parameter in the `wkplc.properties` file of your primary node.
   - Remove all default membership policies and then click **Subexpression builder**. Enter the following information in the Subexpression builder window:
     - Select **Logical operator** pull-down menu. Select **Nodegroup** from the **Select operand** pull-down menu. Select **Equals (=)** from the **Operator** pull-down menu. Type the nodegroup name you created in the previous step in the **Value** text box. Click **Generate subexpression**. Click **Append**. Click **Preview membership** to verify that all nodes included in the nodegroup display and then click **Next**. Click **Create the cluster member using an existing server as a template** radio button and then select the WebSphere_Portal server for the primary node from the pull-down menu.
   - Click **Next**. Specify the dynamic cluster properties and then click **Next**.
   - Review the summary page to verify your actions and then click **Finish**.
5. Define or verify the following parameters in the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory:
   - Set **ClusterName** to the name of the new dynamic cluster.
   - Verify that **CellName** is set to the deployment manager cell.
   - Verify that **NodeName** is set to the local WebSphere Portal node.
   - Set **ServerName** to the server that will be used for the dynamic cluster member on this node.
   - Verify that **PrimaryNode** is set to true.
6. Run the `./ConfigEngine.sh cluster-node-config-dynamic-cluster-setup -DWasPassword=dmgr_password` task from the `wp_profile_root/ConfigEngine` directory to create the dynamic...
13. Install any additional nodes to the cell to support additional cluster members for Cluster B identically to the primary node, and then federate as them as secondary nodes and define as cluster members on these nodes. For information about adding additional nodes navigate to Installing WebSphere Portal > Setting up WebSphere Portal > Setting up a clustered production environment. Select Environment > WebSphere Variables. From the Scope drop-down menu, select the Node=nodename, Server=servername option to narrow the scope of the listed variables, where Node=nodename is the node that contains the application server. Update the WCM_HOST variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router. Update the WCM_PORT variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router. Save your changes and then restart the deployment manager, the node agent(s), server1, and the WebSphere Portal servers. If you are using a Web server to connect to the On Demand Router (ODR), configure the web server as a trusted proxy on the ODR. Refer to Configuring a Web server as a trusted proxy server for instructions. Tip: You can also configure the ODR to dynamically update the Web server configuration when changes occur. Refer to Configuring an on demand router to dynamically update the Web server plug-in configuration for instructions.

Change the task list settings to point to the cluster instead of a single server if using Portal business process integration. See one of the following sections in the Information Center for information: Cross cell: Adding a BPI-enabled portal server to a managed cell in a cross-cell setup Single cell: Adding a BPI-enabled portal server to a managed cell in a single-cell setup

14. Restart the server1 and WebSphere_Portal servers on Cluster A and Cluster B.

15. If you entered passwords in any of the properties files while creating your cluster, you should remove them for security purposes. See “Deleting passwords from properties files” under Related tasks for information.

Installation of Cluster B is complete. It is now an independent cluster from Cluster A, which means that Cluster B can have its own configuration, set of end-user portlets, and target community. Any applications that are common between Cluster A and Cluster B are most likely infrastructure or related to administration, and special care needs to be taken to preserve their commonality between clusters and correct maintenance levels.

Parent topic: Setting up multiple clusters on Linux

Related tasks
Deleting passwords from properties files
Adding a BPI-enabled portal server to a managed cell in a cross-cell setup
Adding a BPI-enabled portal server to a managed cell in a single-cell setup
Routing requests across clusters on Linux

The HTTP Server plug-in that comes with IBM® WebSphere® Application Server is typically used to balance requests for applications across members of the cluster. While an application can be mapped to more than one cluster, automatic plug-in generation does not provide routing or balancing traffic for the same application across multiple clusters. Multiple cluster environments with shared applications therefore cannot rely solely on WebSphere Application Server automatic plug-in generation to be able to route requests using a web server. One option in this scenario is developing a customized method for defining and maintaining the plug-in configuration file used by the Web server to provide for the required routing of user requests.

An important consideration in a multiple cluster environment is ensuring that all subsequent HTTP requests for an end user are routed to the same cluster that processed the first HTTP request. The WebSphere Portal login processing depends upon preserving this cluster affinity during this initial time until the user has successfully logged in and session cookies maintain affinity. In order to guarantee that affinity is preserved during login, set the Navigator Service public.session parameter to a value of true. Refer to “Portal Configuration Services” for information on how to configure this parameter.

Parent topic: Setting up multiple clusters on Linux
Sharing database domains between clusters on Linux

To help provide redundancy and failover support in production environments composed of multiple clusters in a single cell and multiple clusters in different cells, you can share database domains between those clusters. IBM® WebSphere® Portal data is organized into different database domains, with different availability requirements depending on how the production environment is set up. When multiple lines of production are involved and each line of production is implemented as a cluster of servers, sharing database domains ensures that the data is automatically synchronized across the lines of production.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on Linux
  - Preparing your Linux operating system
  - Preparing the primary node on Linux
  - Tune your servers

**Important:** JCR domains and release domains cannot be shared between different Web Content Management clusters or servers. Each distinct cluster or server in your Web Content Management system must use a separate JCR or release domain. For example:

<table>
<thead>
<tr>
<th>Development Server</th>
<th>Authoring Cluster</th>
<th>Staging Server</th>
<th>Delivery Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCR Domain 1</td>
<td>JCR Domain 2</td>
<td>JCR Domain 3</td>
<td>JCR Domain 4</td>
</tr>
<tr>
<td>Release Domain 1</td>
<td>Release Domain 2</td>
<td>Release Domain 3</td>
<td>Release Domain 4</td>
</tr>
</tbody>
</table>

Perform the following steps to share database domains when setting up an environment with multiple clusters:

1. Set up the first cluster (referred to as Cluster A in these instructions).
2. Determine which database domains you want to share with any other clusters in the environment.
3. Install the primary node of the next cluster (Cluster B), and perform the following steps to configure the node to use the shared database domains.
   A. Perform a partial database transfer of the database domains that you are not sharing. For example, if you are sharing only the Customization and Community domains, you would transfer the remaining domains to the database you are using for Cluster B.
   B. Reconfigure the shared database domains on the node to connect to the shared database domains you are using for Cluster A. For example, to connect to the Customization and Community domains for Cluster A, you would connect to those domains from Cluster B.
4. Continue setting up the primary node as described in the cluster instructions.
5. Install the secondary node of Cluster B, and perform the following steps to configure the node to use the shared database domains.
   A. For those database domains that you are not sharing between clusters, reconfigure the domains to connect to the database domains you are using for Cluster B. As in the example for the primary node, if you are sharing only the Customization and Community domains, reconfigure the remaining domains on the secondary node to use the domains of the primary node for Cluster B.
   B. Reconfigure the shared database domains on the node to connect to the shared database domains you are using for Cluster A. For example, to connect to the Customization and Community domains for Cluster A, you would connect to...
those domains from Cluster B.

6. Continue setting up the secondary node as described in the cluster instructions.

**Parent topic:** Setting up a cluster on Linux  
**Previous topic:** Setting up multiple clusters on Linux  

**Related tasks**  
Connecting to existing database domains
Setting up a cluster on Solaris

In a production environment, you can install and configure IBM® WebSphere® Portal on a single (primary) node and then add additional nodes in the future as your company's needs grow. You can also deploy a high availability production environment by creating and configuring clustered deployments of multiple servers running WebSphere Portal managed by IBM WebSphere Application Server Network Deployment.

- **Prerequisites**
  - [Technotes for installation and configuration issues](#)

Perform the following steps to set up your production environment on Solaris:

1. **Preparing prerequisite and corequisite software on Solaris**
   Before creating your high availability production environment, you must prepare your database, your user registry, your deployment manager, and your remote Web server.

2. **Preparing your Solaris operating system**
   View information on setting up your operating system for IBM WebSphere Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

3. **Preparing the primary node on Solaris**
   Before creating your high availability environment, you must install IBM WebSphere Portal on your primary node and then configure your database and your network deployment manager.

4. **Choosing the type of cluster to create on Solaris**
   If you installed a IBM WebSphere Application Server Network Deployment, create a static cluster to handle failover requests. If you installed a IBM WebSphere Virtual Enterprise, create a dynamic cluster to balance member workloads.

5. **Preparing a remote Web server when portal is installed on Solaris**
   Install and configure the Web server plug-in, provided by IBM WebSphere Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

6. **Configuring WebSphere Portal to use a user registry on Solaris in a clustered environment**
   Configure user registry security on IBM WebSphere Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

7. **Preparing additional nodes on Solaris**
   After installing and configuring your primary node, you can create your secondary nodes. You must install IBM WebSphere Portal on each node and then configure the node to access the database and user registry before adding it to the cluster.

8. **Tune your servers**
   Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

9. **Configuring search in a cluster on Solaris**
   IBM WebSphere Portal provides two distinct search capabilities. You can utilize both types of search capabilities in a clustered environment.
10. **Setting up multiple clusters on Solaris**

The majority of the steps to build another cluster are the same as when building the first cluster in the same cell, with a few exceptions. Basically, the new profile will be designated as the primary profile, using IBM WebSphere Portal clustering terminology, and will then be used as the basis for the new cluster definition. This duplicates the build process of the first cluster in the cell. During the federation process, if any applications on this new node already exist in the cell (because they are in use by the first cluster), then Deployment Manager will not allow them to be added. After federation, the applications that already exist in the cell are not mapped to the WebSphere_Portal server on the newly federated node, and thus the existing applications must be remapped to this newly federated server to restore its application list. Therefore, depending on the configuration of the new profile, there will likely be some combination of applications shared with the other existing cluster, and some applications unique to this new profile.

11. **Sharing database domains between clusters on Solaris**

To help provide redundancy and failover support in production environments composed of multiple clusters in a single cell and multiple clusters in different cells, you can share database domains between those clusters. IBM WebSphere Portal data is organized into different database domains, with different availability requirements depending on how the production environment is set up. When multiple lines of production are involved and each line of production is implemented as a cluster of servers, sharing database domains ensures that the data is automatically synchronized across the lines of production.

**Parent topic:** Setting up a cluster
Preparing prerequisite and corequisite software on Solaris

Before creating your high availability production environment, you must prepare your database, your user registry, your deployment manager, and your remote Web server.

Perform the following tasks to prepare prerequisite and corequisite software:

1. **Preparing the WebSphere Application Server Deployment Manager on Solaris**
   IBM WebSphere Portal provides a customized installation package (CIP) that includes IBM WebSphere Application Server Network Deployment and all required maintenance packages. Use the supplied discs to install WebSphere Application Server Network Deployment on a dedicated system.

2. **Preparing a remote Web server when portal is installed on Solaris**
   Install and configure the Web server plug-in, provided by IBM WebSphere Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

**Parent topic:** Setting up a cluster on Solaris

**Next topic:** Preparing your Solaris operating system
Preparing the WebSphere Application Server Deployment Manager on Solaris

IBM® WebSphere® Portal provides a customized installation package (CIP) that includes IBM WebSphere Application Server Network Deployment and all required maintenance packages. Use the supplied discs to install WebSphere Application Server Network Deployment on a dedicated system.

Perform the following steps to prepare the WebSphere Application Server Deployment Manager:

1. Install WebSphere Application Server Network Deployment or use the CIP to upgrade an existing installation. Run the following command: cd_root/Solaris/architecture/ifpackage/NAS/install, where cd_root is the root directory of the disc and architecture is the system's processor architecture.

2. Choose one of the following options to create a default deployment manager profile: Important: While creating the default deployment manager profile, enable administrative security. If you use the Profile Management Tool, check the enable administrative security check box. If you use the manageprofile command, add the -enableAdminSecurity=true parameter to the command line.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile Management Tool</td>
<td>See Creating a deployment manager profile for information.</td>
</tr>
<tr>
<td>manageprofile commands</td>
<td>See manageprofile commands for information. Note: If you have a 64-bit environment, only the manageprofiles command is supported when creating profiles.</td>
</tr>
</tbody>
</table>

3. Run the following command to start the deployment manager: ./startManager.sh, from the dmgr_profile_root/bin directory.

4. Use the following URL to launch the network deployment administrative console: http://dmgr_hostname:9060/ibm/console, where dmgr_hostname is the fully qualified host name for the WebSphere Application Server Network Deployment.

5. Log into the deployment manager administrative console.

6. Increase the HTTP connection timeouts for the deployment manager:

   A. Click System Administration > Deployment Manager > Web container transport chains.
   
   B. Increase the timeout values. For the WCInboundAdmin and WCInboundAdminSecure entries listed in the web container transport chains section, complete the following steps to increase the timeout values:

      1. Click HTTP Inbound Channel.
      2. Change the Read timeout value to 180.
      3. Change the Write timeout value to 180.
      4. Save the configuration changes.

7. Change the timeout request period for the Java Management Extensions (JMX) connector.

   A. Click System administration > Deployment Manager > Administration Services > JMX connectors > SOAPConnector > Custom Properties.
   
   B. Select the requestTimeout property, and increase the value from 600 to 6000.
   
   C. Save the configuration changes.

8. Update the maximum Java heap size used by the deployment manager:

B. Update the value in the **Maximum Heap Size** field. For information about appropriate heap sizes see the documentation for your operating system in the Performance Guides located on the WebSphere Portal and Web Content Management Product Documentation page. **Note:** If using a 32-bit operating system, you will need to set the heap size to a lower size than a 64-bit operating system.

C. Click **OK** and then save your changes.

9. Depending on your version of WebSphere Application Server, make the appropriate selection from the navigation:
   - **Note:** If security is not enabled on your deployment manager, see **Enabling security** for information before performing this step.
     - For **WebSphere Application Server Version 6.1**: Click **Security > Secure administration, applications and infrastructure** and select **Enable Application Security**. Then save the configuration changes.
     - For **WebSphere Application Server Version 7.0**: Click **Security > Global security** and select **Enable Application Security**. Then save the configuration changes.

10. Verify that the WebSphere Portal administrative users and administrative group exist in the Deployment Manager cell's user registry. Perform the following steps if you need to create the administrative users and group:
   A. Click **Users and Groups > Manage Users**.
   B. Click **Create**.
   C. Type the information for the WebSphere Portal administrative users; for example **wpsadmin** and **wpsbind**, and then click **Create**.
   D. Click **Users and Groups > Manage Groups**.
   E. Click **Create**.
   F. Type **wpsadmins** as the name of the WebSphere Portal administrative group and then click **Create**.
   G. Click the group you just created; for example **wpsadmins**.
   H. Click the **Members** tab.
   I. Click **Add Users**.
   J. Search for the users.
   K. Select the users you want to add to the group.
   L. Click **Add** to add the users to the group.
   M. Click **Close** when you are done adding users to the group.
   N. Log out of the administrative console.

11. Change the timeout request period for the Simple Object Access Protocol (SOAP) client. Edit the **soap.client.props** file, located in the Dmgr_profile/properties directory: Change the line to: **com.ibm.SOAP.requestTimeout=6000**.

12. Run the following tasks to stop and restart the deployment manager:
   A. **./stopManager.sh -username admin_userid -password admin_password**, from the dmgr_profile_root\bin directory
   B. **./startManager.sh**, from the dmgr_profile_root\bin directory

**Parent topic:** Preparing prerequisite and corequisite software on Solaris

**Next topic:** Preparing a remote Web server when portal is installed on Solaris
Preparing a remote Web server when portal is installed on Solaris

Install and configure the Web server plug-in, provided by IBM® WebSphere® Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

**Prerequisites**
- Preparing the WebSphere Application Server Deployment Manager on Solaris
- Preparing prerequisite and corequisite software on Solaris
- Preparing your Solaris operating system
- Preparing the primary node on Solaris

Perform the following steps to install and configure your Web server:

1. Install and configure the Web server; refer to the Web server documentation for information.
2. If using IBM Lotus® Domino®, edit the **NOTES.INI** file on the Web server. Set the **HTTPEnableConnectorHeaders** parameter to 1.
3. If using IBM HTTP Server or Apache Server, edit the **httpd.conf** file on the Web server. Set the **AllowEncodedSlashes** directive to **On**; the directive should be added at the root level as a global directive.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>See the appropriate HTTP Server</td>
<td>IBM HTTP Server</td>
</tr>
<tr>
<td>documentation</td>
<td></td>
</tr>
<tr>
<td>See the appropriate Apache Server</td>
<td>AllowEncodedSlashes</td>
</tr>
<tr>
<td>documentation</td>
<td>directives</td>
</tr>
</tbody>
</table>

4. Stop the Web server.
5. Install and configure the Web server plug-in on the system where the Web server is located using the Plug-ins installation wizard provided with WebSphere Application Server. Refer to [Selecting a Web server topology diagram and roadmap](#) for information. **Note:** The WebSphere Application Server Web server instructions are for the default profile. If not using the default profile, refer to [Configuring a Web server for a non-default profile](#).

   If using WebDAV: After successfully installing the Web server plug-in, locate and open your **plugin-cfg.xml** file and set **AcceptAllContent** to **true**.

   **Important:** Depending on how you use the Web server, you may need to adjust the **ServerIOTimeout** value, which defines how long the plug-in should wait for a response from the application. The recommended minimum value is 60 but you may need to adjust this value higher if you are retrieving data from a database. To update this value, locate and open your **plugin-cfg.xml** file and set **ServerIOTimeout** to a value that is appropriate for your business needs. For additional information, see [Common questions about the Web server plug-in](#).

6. If you are using Sun Web Server Version 7 update 8, read Technote 1448262 and perform the steps to resolve the HTTP 408/409 error.
7. Start the Web server.
Preparing your Solaris operating system

View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on Solaris

Perform the following steps to prepare your operating system:

1. Perform the following steps to prepare your Solaris kernel settings: Several Solaris kernel values are typically too small for the messaging requirements of WebSphere Portal. Starting the internal JMS server or client with insufficient kernel resources produces a First Failure Support Technology (FFST) file in the /var/errors directory. Before installing WebSphere Portal, review the machine's configuration.

   **Note:** The values described here are a starting point for messaging in WebSphere Portal only. If your system has other applications installed, the value requirements will likely be different. For example, if values that are already set are higher than the settings listed here, the values should not be lowered. Be sure to check the requirements made on /etc/system by other already-installed applications before altering existing values.

   A. Type the `sysdef -i` command to review the configuration.
   
   B. Set `shmsys:shminfo_shmmax` (valid for Solaris Version 9 only) to **4294967295**.
   
   C. Set `shmsys:shminfo_shmni` (valid for Solaris Version 9 only) to **1024**.
   
   D. Set `semsys:seminfo_semaem` (valid for Solaris Version 9 only) to **16384**.
   
   E. Set `semsys:seminfo_semni` (valid for Solaris Version 9 only) to **1024**.
   
   F. Set `semsys:seminfo_semmns` (valid for Solaris Version 9 only) to **16384**.
   
   G. Set `semsys:seminfo_semmsl` (valid for Solaris Version 9 only) to **100**.
   
   H. Set `semsys:seminfo_semmopm` (valid for Solaris Version 9 only) to **100**.
   
   I. Set `semsys:seminfo_semmnu` (valid for Solaris Version 9 only) to **2048**.
   
   J. Set `semsys:seminfo_semmue` (valid for Solaris Version 9 only) to **256**.
   
   K. Set `msgsys:msginfo_msgmap` (valid for Solaris Version 9 only) to **1024**.
   
   L. Set `msgsys:msginfo_msgmax` (valid for Solaris Version 9 only) to **65535**.
   
   M. Set `rlim_fd_cur` to **1024**.

2. **Web Content Management only:** Use the `ulimit -f` command to set the maximum size of files that can be created to be at least the size of the largest file you would need to upload to the content server. The command `ulimit -f -1` removes any limit on file size.

3. Set the file descriptor limit to **10240**; for example, `ulimit -n 10240`.

4. Perform the following steps to prepare for non-global zone:

   A. Do not inherit package directories when you create the non-global zone because the inherited software packages will be read-only.

   B. Stop WebSphere Portal and all related processes before installing or uninstalling.

   C. Verify that the following processes are stopped:

      - `/opt/IBM/WebSphere/AppServer/java/bin/java`
      - `/opt/IBM/WebSphere/AppServer/java/jre/bin/java`

---

**Parent topic:** Setting up a cluster on Solaris

**Previous topic:** Preparing prerequisite and corequisite software on Solaris
Next topic: Preparing the primary node on Solaris
Preparing the primary node on Solaris

Before creating your high availability environment, you must install IBM® WebSphere® Portal on your primary node and then configure your database and your network deployment manager.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on Solaris
  - Preparing your Solaris operating system

Perform the following tasks to prepare your primary node:

1. **Installing WebSphere Portal on Solaris on the primary node**
   IBM WebSphere Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

2. **Configure WebSphere Portal to use a remote database**
   For high availability, using a remote database is preferred. For improved performance databases can be distributed across multiple database servers. Databases should also be configured for failover. Configuring the database for failover is beyond the scope of this documentation. Refer to the database product documentation for the most authoritative guidance about setting up databases for fail over.

3. **Configuring the primary node to communicate with the deployment manager on Solaris**
   After installing IBM WebSphere Application Server Network Deployment and installing IBM WebSphere Portal on the primary node, you must configure the primary node and the deployment manager to communicate with each other.

4. **Removing search collections on Solaris**
   If you plan to use search in a cluster, you must configure a remote search server; see Configuring Portal Search in a cluster for information. If you created any search collections, you will need to recreate them on the remote search server. If your search collection has data, export the collection before deleting it and then import the data on the remote server.

**Parent topic:** Setting up a cluster on Solaris

**Previous topic:** Preparing your Solaris operating system

**Next topic:** Choosing the type of cluster to create on Solaris
Installing WebSphere Portal on Solaris on the primary node

IBM® WebSphere® Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

Review the Supported hardware and software requirements for your operating system and ensure that all hardware and software matches the minimum product level before installing WebSphere Portal. Review Installation methods, options, and sources before installing to determine your installation source location and installation method.

Perform the following steps to install WebSphere Portal and create a custom profile for the node that will be added to the cluster. If you already installed WebSphere Portal, you can skip to the last step to create the custom profile.

1. Type `ping yourserver.yourcompany.com` on a command line to verify that your fully qualified host name is properly configured.
2. Type `ping localhost` on a command line to verify that your network is properly configured.
3. If you are installing with an existing WebSphere Application Server instance, ensure it is installed at the supported level and has the following features installed:
   - Application Server
   - Administration
   - Scripted Administration
   - Administrative Console
   - Ant and Deployment Tools
   - Deploy Tool
   - Ant Utilities
   **Restriction:** Installing WebSphere Portal Version 6.1.0.1 or higher on an existing WebSphere Application Server Version 7.0 instance where WebSphere Portal Version 7.0 was installed is unsupported.
   **Important:** Besides ensuring that the existing WebSphere Application Server instance is at the supported level, you will also need to ensure that Apache Derby is installed at the supported level before installing WebSphere Portal. See WebSphere Portal detailed system requirements for supported levels.

4. Optional: Perform the following steps if you want to install WebSphere Portal using a non-root user:
   **Restriction:** Although it is possible to install WebSphere Application Server as a non-root user, there are limitations to this option that can affect WebSphere Portal functionality, which is why you should install WebSphere Application Server as a root user.
   A. Install the current supported version of WebSphere Application Server as a root user.
   B. Open a command prompt and perform the following steps to create a non-root user and to change ownership:

   1. Use the appropriate system commands to create a new group, to create a new user, and to add the new user to the new group.
      
      ```
      chgrp -R group_name/opt/IBM
      chmod -R g+rw /opt/IBM
      chmod -R g+wr /tmp
      ```
   2. Run the following tasks to change the rights of the non-root user:

      ```
      chgrp -R group_name/opt/IBM
      chmod -R g+rw /opt/IBM
      chmod -R g+wr /tmp
      ```
chgrp -R group_name /tmp
chmod -R g+wr /var/tmp
chgrp -R group_name /var/tmp

C. Launch the installation program per the next step. During the installation, select the existing WebSphere Application Server using the non-root user and set the installation location to a unique location under the path where non-root permissions were granted.

5. Choose one of the following installation commands based on the installation method you want to use to install WebSphere Portal; see Installation Methods for more information:

**Advance Installation parameters:** To customize your installation, you can add various parameters to your installation commands; for example, you can specify your port information. See Advanced installation parameters for information.

**Restriction:** When defining your installation path, the **ONLY** supported characters are ASCII letters [A-Z and a-z], numbers [0-9], slashes [/ or \, depending on your operating system], and the dash [-].

**Table 1. Installation task options**

<table>
<thead>
<tr>
<th>Installation type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphical user interface</td>
<td>./install.sh</td>
</tr>
<tr>
<td>Console mode</td>
<td>./install.sh -console from the PortalExpress subdirectory</td>
</tr>
<tr>
<td>Silent install</td>
<td>./install.sh -options &quot;path_to_file/response_filename&quot;, where path_to_file is the full path to the response file and response_filename is the name of the file. A sample install response file (installresponse.txt) and a sample uninstall response file (uninstallresponse.txt) are located in the setup CD root directory. <strong>Important:</strong> Do not place the response file in a path that contains a space and do not put a space in the file name.</td>
</tr>
</tbody>
</table>

**Important:** If you have a Solaris 9 on Sun SPARC, you MUST either add the -W defaults.force32bit=true parameter to your installation command or install the 32-bit WebSphere Application Server product before installing WebSphere Portal to prevent the installer from automatically selecting a 64-bit environment.

**Note:** If the installation program does not detect a WebSphere Application Server instance that you know exists, exit the installation program and pass the WebSphere Application Server instance location using the command line; for example, ./install.sh -W was.undetectedWas="/my/WAS/location".

**Note:** If the GUI or console mode installation program fails to detect ports for either WebSphere Application Server or WebSphere Portal, a warning message displays and the installer offers another chance to enter the values. If the silent installation fails to detect ports for either WebSphere Application Server or WebSphere Portal, the installer will exit.

6. Verify your installation was successful; access WebSphere Portal using the [http://yourserver:yourport/wps/portal format](http://yourserver:yourport/wps/portal). Run the ./ConfigEngine.sh list-server-ports -DWasPassword=password task from the wp_profile_root/ConfigEngine directory to generate the wp_profile_root/ConfigEngine/log/wp_PortMatrix.txt file that lists WebSphere Portal ports for your installation.

7. Optional: After the WebSphere Portal installation completes successfully, run the ./ConfigEngine.sh configure-express -DPortalAdminPwD=password -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, if you want to enable the sample IBM Lotus Web Content Management content, such as internet and intranet sample sites.

**Restriction:** Run the configure-express task before configuring your database, user registry, context root, security, etc. If you ran any tasks other than the install task, do not run this task.

**Note:** See [http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_ovw.html](http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_ovw.html)
for tutorials on how to use the sample content.

The sample content includes:

- Creates a group called contentAuthors; members of this group are given privileges to create content in the sample Internet and intranet sites. Navigate to the **Administration** area and then click **Access > Users and Groups**.
- Creates two new Web Content Management Libraries: "Internet Web Content 6.1.0" and "Intranet Web Content 6.1.0". Navigate to the **Administration** area and then click **Portal Content > Web Content Libraries**.
- Adds a portlet filter and applies the filter to various portlets in the sample Internet and intranet sites. You can see the definition of the filter in the WebSphere Application Server Administration console and examining the custom resources under the **Environment** area.
- Creates two new theme policies: InternetStyle and IntranetStyle. These styles are applied to sample Internet and intranet sites. Navigate to **Theme Customizer** and then select the style.
- Creates several portlet clones of the Web Content Management rendering portlet. These portlet clones are used on sample Internet and intranet sites.
- Creates two virtual portals with context roots of wps/portal/intranet and wps/portal/internet. These are the sample Internet and intranet sites. Go to http://yourserver:yourport/wps/portal/internet and http://yourserver:yourport/wps/portal/intranet to access them.
- Creates several sample credential slots, including "Default slot for E-mail", "Default slot for Feeds", "Default slot for Miscellaneous", "Default slot for Web Clipping", and "Default slot for Web Content Management". Navigate to the **Administration** area and then click **Access > Credential Vault > Manage System Vault Slots**.
- Modifies the portlet palette to include some new portlets. Click the **Portlets** link to see the portlet palette.

8. Optional: If you ran the configure-express task, the owner of the items in the Web content libraries containing the Internet and Intranet Site Template content will be listed as uid=xyzadmin,o=defaultWIMFileBasedRealm. To update the owner information for these items to correspond to your portal administrator ID, complete the following steps:

A. Edit the wp_profile_root
   /PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties file.

B. Add the following line to the file:
   ```
   uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN
   ```
   Replace `portal_admin_DN` with the distinguished name of the portal administrator.

C. Save your changes and close the file.

D. Run the `./ConfigEngine.sh action-express-memberfixer -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory.

9. Optional: Perform the following steps if you need to change the ports for WebSphere Application Server or WebSphere Portal:

   **Note:** The starting port parameter is required for a successful completion of the modify-ports-by-startport task. Once you specify a start port, this port becomes the base for assigning port values. The code increments this value as each port is assigned, which means that the WebSphere Portal ports will be assigned incrementally starting with the port defined with the modify-ports-by-startport task.

A. Stop the server1 and WebSphere_Portal servers.
B. Run one of the following commands for each server you need to change:

<table>
<thead>
<tr>
<th>Method</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting port number</td>
<td>./ConfigEngine.sh modify-ports-by-startport -DWasPassword=password -DModifyPortsServer=servername -DStartPort=start_port_number</td>
</tr>
</tbody>
</table>
C. Restart the server1 and WebSphere_Portal servers.

10. To prevent Out of Memory errors, perform the following steps to set the MaxPermSize:

Tip: If Maximum Heap Size is set to 2048 M or higher, set the MaxPermSize to a quarter of the value entered for the Maximum Heap Size; for example, if your Maximum Heap Size is 3000 M, set MaxPermSize to 750 M. If your Maximum Heap Size is less than 2048 M, set MaxPermSize to 512 M.

A. Log in to the WebSphere Application Server Administration Console.

B. Click Servers > Server Types > WebSphere application servers > WebSphere_Portal.


D. In the Generic JVM arguments field, change the MaxPermSize value to -XX:MaxPermSize=numeric value, where numeric value is a quarter of the value entered for the Maximum Heap Size. Important: If MaxPermSize does not exist in the Generic JVM arguments field, add it to the field but do not replace existing information in the Generic JVM arguments field with the MaxPermSize information.

E. Click OK to save your changes.

F. Click Save to save your changes to the master configuration.

G. Log out of the WebSphere Application Server Administration Console.

H. Restart the server1 and WebSphere_Portal servers.

Parent topic: Preparing the primary node on Solaris

Next topic: Configure WebSphere Portal to use a remote database

Related concepts
Installation methods
IBM Support Assistant Lite for WebSphere Portal

Related reference
Advanced installation parameters
Configure WebSphere Portal to use a remote database

For high availability, using a remote database is preferred. For improved performance databases can be distributed across multiple database servers. Databases should also be configured for failover. Configuring the database for failover is beyond the scope of this documentation. Refer to the database product documentation for the most authoritative guidance about setting up databases for fail over.

Password considerations when transferring data manually

For security reasons, you should not store passwords in the wkplc.properties and wkplc_comp.properties. Edit each of the properties files prior to running a configuration task, inserting the passwords needed for that task. After the task has run, delete all passwords from each file. For more information, see Deleting passwords from configuration scripts. Alternatively, you can specify the password on the command line using the following syntax:

- Windows: ConfigEngine.bat task_name -Dpassword_property_key=password_value
- UNIX: ./ConfigEngine.sh task_name -Dpassword_property_key=password_value
- i5/OS: ConfigEngine.sh task_name -Dpassword_property_key=password_value

As with other properties, each password property must have the -D prefix and be set equal to (=) a value. If you have multiple properties in a single command, use a space character between each -Dproperty=value setting.

- Prerequisites
  - Installing WebSphere Portal on Solaris on the primary node
  - Technotes for database connectivity issues

- Configuring WebSphere Portal to use DB2
  View information on installing and setting up DB2 to work with WebSphere Portal.

- Configuring WebSphere Portal to use DB2 for z/OS
  View information on installing and setting up DB2 for z/OS to work with WebSphere Portal.

- Configuring WebSphere Portal to use Oracle
  View information on installing and setting up Oracle to work with WebSphere Portal.

- Configuring WebSphere Portal to use Oracle RAC
  View information on installing and setting up Oracle RAC to work with WebSphere Portal.

- Configuring WebSphere Portal to use SQL Server
  View information on installing and setting up SQL Server to work with WebSphere Portal.

- Verifying databases
  After you configure IBM WebSphere Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.

Parent topic: Preparing the primary node on Solaris
Previous topic: Installing WebSphere Portal on Solaris on the primary node
Next topic: Configuring the primary node to communicate with the deployment manager on Solaris

Related Information
Configuring WebSphere Portal to use DB2

View information on installing and setting up DB2 to work with WebSphere Portal.

1. **Installing DB2**
   View information on installing DB2 for use with WebSphere Portal.

2. **Create users**
   View the steps to set up users for DB2 to work with WebSphere Portal.

3. **Creating remote databases**
   A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

4. **Optional: Assigning custom table spaces**
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

5. **Modifying database properties**
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

6. **Optional: Configuring JCR collation support**
   View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

7. **Configure WebSphere Portal to use DB2**
   View the steps to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

8. **Configuring DB2 for large file handling in Web Content Management**
   If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

9. **Optional: Changing driver types**
   WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

**Parent topic:** Configure WebSphere Portal to use a remote database
Installing DB2

View information on installing DB2 for use with WebSphere Portal.

Before you begin:

- Make sure DB2 is set up with updated kernel parameters according to the recommendations in the DB2 Quick Beginnings guide at DB2 Technical Support.

- **(For Type 2 drivers only):** DB2 Connect must be installed on the same system as WebSphere Portal, and have the same name as the server profile name.

- When you install DB2 using the DB2 installation program, it automatically creates a DB2 administrative user with the correct operating system rights.

1. If you are using the JDBC driver in type 2 mode, configure your DB2 Connect client with the following commands. If you are using a remote database, complete this step separately from the WebSphere Portal installation.
   - `db2 update dbm cfg using tp_mon_name WAS`
   - `db2 update dbm cfg using spm_name hostname`, where `hostname` is the host name of WebSphere Portal.

   Because the default for `spm_name` is the hostname itself, specifying the `hostname` parameter is optional. If your hostname is more than eight characters, use empty double quotes (" "). For example, `db2 update dbm cfg using spm_name " "`.

2. Before installing DB2, log in with a user ID that has administrative authority.

3. To install DB2, follow the instructions that are provided with the DB2 documentation to install DB2 and the required fix pack.

**Parent topic:** Configuring WebSphere Portal to use DB2

**Next topic:** Create users

**Related Information**

- DB2 Technical Support
- DB2 Information Center
Create users

View the steps to set up users for DB2 to work with WebSphere Portal.

Before you begin: You should have completed Installing DB2.

- **Prerequisites**
  - Installing DB2

For all databases, use one user with administrative rights on the operating system and the DB2 installation. This user can be the database administrative user that is created automatically by the DB2 installation program. This user is the database configuration user that will be used for configuration tasks: creating database tables and performing database transfer. If you choose to have WebSphere Portal also create the databases, the database user should be given `SYSADM` rights. You only need to manually create an administrator ID when you do not want to use an existing DB2 administrator ID.

A common user name is `db2inst1`, but you can assign any user name as long as it has administrative access and follows the limitations listed here. Do not change the user name after creating it.

The limitations on user names are:

- User names can contain one to eight characters.
- Group and instance names can contain one to eight characters.
- Names cannot be any of the following: `users` `admins` `guests` `public` `local`
- Names cannot begin with: `IBM` `SQL` `SYS`
- Names cannot include accented characters.
- Create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Following are the basic steps to create new users and grant privileges. For more information, refer to the DB2 product documentation.

1. From the DB2 Control Center, expand the object tree until you find the DB Users folder.
2. Right-click the DB Users folder and select **Create/Add**. The Create DB User notebook opens.
3. In the User name field, type a user ID.
4. Assign an authority for the user.
5. Click **OK**.

**Parent topic:** Configuring WebSphere Portal to use DB2

**Previous topic:** Installing DB2

**Next topic:** Creating remote databases
Creating remote databases

A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

Before creating these databases, note the following information:

- Use IBM® DB2 Universal Database™ Enterprise Server Edition JDBC driver, JCC, type 4.

- If the DB2 JDBC type 4 driver is used, only the instructions on the remote server are required. You do not need to install the DB2 client software, and you do not need to complete the steps related to the DB2 client.

- The client software, DB2 Connect, must be correctly configured to connect to the remote DB2 server instance, for example, db2inst1.

- These instructions assume that a remote DB2 server and DB2 Connect are already installed and running. For information on installing DB2 Connect, see the DB2 product documentation.

- **Prerequisites**
  - Installing DB2
  - Create users

To create a database, you must be a DB2 System Administrator with sufficient database privileges (SYSADM or at a minimum SYSCtrl).

1. Log in to the DB2 server system as a user with sufficient database privileges.
2. Initialize a DB2 command environment by opening a command prompt and typing `su - db2inst1`, where `db2inst1` is the database user. The prompt changes to your operating system shell prompt, for example: $.
   In this mode, you must type `db2` at the beginning of each command and use double quotation marks (""") around the entire command. The following example shows how to use the double quotes; it is not an actual command that you run:
   ```
   db2 "CREATE REGULAR TABLESPACE SMS PAGESIZE 4 K MANAGED BY SYSTEM USING ('sms') BUFFERPOOL SMSPOOL"
   ```

   **Note:** If you are using the Command Line Processor (CLP), refer to your DB2 documentation for details. The command prompt is `db2=>`. In this mode, commands can be entered without the `db2` prefix or the double quotation marks.

   However, the following steps assume you are not using the CLP and are entering commands from your operating system shell prompt, for example: $.

3. Run the following commands on the DB2 server system to configure the DB2 database instance:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Description</th>
</tr>
</thead>
</table>
   | **DB2**     | db2set DB2_RA_TO_RS=YES
                | DB2_EVALUNCOMMITTED=YES
                | dbset DB2_INLIST_TO_NLJN=YES
                | db2 "UPDATE DBM CFG USING query_heap_sz 32768"
                | db2 "UPDATE DBM CFG USING maxagents 500"
                | db2 "UPDATE DBM CFG USING sheapthres 0"

4. **(Important)** Failure to complete this step will result in an unsuccessful database transfer. Run the following commands on the DB2 server system to create the necessary databases:

   - Replace `dbname` with the actual name of the database. Run the commands and each time replace `dbname` with the actual values for release, community, customization, Java Content Repository, Feedback, and Likeminds. You will need to run the commands once for each database for a total of six times.

   **Remember:** DB2 database names cannot
exceed eight characters. Therefore, consider using these database names: release, commun, custom, jcrdb, fdbkdb, and lmdb.

```
db2 "CREATE DB dbname using codeset UTF-8 territory us PAGESIZE 8192"
db2 "UPDATE DB CFG FOR dbname USING applheapsz 4096"
db2 "UPDATE DB CFG FOR dbname USING app_ctl_heap_sz 1024"
db2 "UPDATE DB CFG FOR dbname USING stmtheap 32768"
db2 "UPDATE DB CFG FOR dbname USING dbheap 2400"
db2 "UPDATE DB CFG FOR dbname USING locklist 1000"
db2 "UPDATE DB CFG FOR dbname USING logfilsiz 4000"
db2 "UPDATE DB CFG FOR dbname USING logprimary 12"
db2 "UPDATE DB CFG FOR dbname USING logsecond 20"
db2 "UPDATE DB CFG FOR dbname USING logbufsz 32"
db2 "UPDATE DB CFG FOR dbname USING avg_appls 5"
db2 "UPDATE DB CFG FOR dbname using AUTO_MAINT off"
```

Complete the following:

5. On the DB2 server system, run the following commands. This step is only required for the IBM Java Content Repository database (jcrdb).

- `jcrdb` is the name of the database used to store user data and objects
- `jcr` is the jcr user for `jcrdb`. Note: This value can be replaced with any ID that has administrative authority.
- `dbpassword` is the password for the jcr user for the jcrdb

```
db2 "CONNECT TO jcrdb USER jcr USING dbpassword"
db2 "CREATE BUFFERPOOL ICMLSFREQBP4 SIZE 1000 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL ICMLSVOLATILEBP4 SIZE 16000 PAGESIZE 4 K"
db2 "CREATE BUFFERPOOL ICMLSMAINBP32 SIZE 16000 PAGESIZE 32 K"
db2 "CREATE BUFFERPOOL CMBMAIN4 SIZE 1000 PAGESIZE 4 K"
db2 "CREATE REGULAR TABLESPACE ICMLFQ32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLFQ32') BUFFERPOOL ICMLSMAINBP32"
db2 "CREATE REGULAR TABLESPACE ICMLNF32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('ICMLNF32') BUFFERPOOL ICMLSMAINBP32"
db2 "CREATE REGULAR TABLESPACE ICMVFQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('ICMVFQ04') BUFFERPOOL ICMLSVOLATILEBP4"
db2 "CREATE REGULAR TABLESPACE ICMSFQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('ICMSFQ04') BUFFERPOOL ICMLSFREQBP4"
db2 "CREATE REGULAR TABLESPACE CMBINV04 PAGESIZE 4 K MANAGED BY SYSTEM USING ('CMBINV04') BUFFERPOOL CMBMAIN4"
```

For JDBC Type 2 connections only: On DB2 Connect, use a text editor to open the `/etc/services` file. If it does not specify the DB2 connection service port, add the following text to specify the port for the remote DB2 instance:
```
db2c_ db2inst1:port1/tcp # DB2 connection service port
```
where `db2inst1` is the name of the DB2 instance ID on the system, and `port1` with the actual port number that is
assigned to the DB2 connection service in your DB2 server installation. The connection service port on the DB2 Client system, WebSphere Portal server, must match the connection service port on the DB2 server. The ports should match by number but not necessarily by name.

C. Set up the correct service name by entering the following command on the DB2 server system:

```
db2 "UPDATE DBM CFG USING svcename svce_name" where svce_name is the connection service port name that is specified above.
```

D. For JDBC Type 2 connections only: On DB2 Connect, set DB2COMM to TCP/IP by using the db2set command

```
db2set DB2COMM=tcpip.
```

E. For JDBC Type 2 connections only: Catalog the TCP/IP node with the IP address of the remote database server on DB2 Connect:

```
db2 "catalog tcpip node remote_db_node_alias remote database_server_node server connection_service_port" where:
- remote_db_node_alias is the alias name of the database server that you are defining for the WebSphere Application Server node name. The alias name can contain one to eight characters.
- database_server_node is the fully qualified host name of your database server system.
- connection_service_port is the name of the DB2 connection service port that is configured in the /etc/services file on the database server system.
```

F. For JDBC Type 2 connections only, catalog the WebSphere Portal databases on DB2 Connect, where:

- remote_db_name_domain, is the cataloged name of the databases on the server system for each domain.
- domain_alias_name, is the database alias names that you are defining.
- remote_db_node_alias is the name that was used previously when you cataloged the TCP/IP node in the previous step.

The alias for each database must be different from the actual database name and can only contain up to eight characters.

```
db2 "catalog db remote_db_name_release as release_alias_name at node remote_db_node_alias"
db2 "catalog db remote_db_name_community as comm_alias_name at node remote_db_node_alias"
db2 "catalog db remote_db_name_customization as cust_alias_name at node remote_db_node_alias"

db2 "catalog db remote_db_name_fdbkdb as fdbkdb_alias_name at node remote_db_node_alias"

db2 "catalog db remote_db_name_lmdb as lmdb_alias_name at node remote_db_node_alias"

db2 "catalog db remote_db_name_jcrdb as jcrdb_alias_name at node remote_db_node_alias"
```

G. For JDBC Type 2 connections only: Log out of DB2 Connect by entering `db2 "terminate"`.

H. For JDBC Type 2 connections only, on DB2 Connect, test your remote connection by issuing the following command in the DB2 command window:

```
db2 "connect to alias_name user username using password", where alias_name is the alias name that you defined above, username is the database user, and password is the password assigned to the database user.
```

**Parent topic:** Configuring WebSphere Portal to use DB2  
**Previous topic:** Create users  
**Next topic:** Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties` file.
- The page size of table spaces used by WebSphere Portal must be 8192 bytes.
- For details on creating table spaces refer to the documentation for the database.

**Prerequisites**
- Installing DB2
- Create users
- Creating remote databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties` that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`
   
   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds

3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword `IN` and a space. For example: `community.COMP_INST.tablespace=IN COMM8KSPACE`
   
   Repeat this step for each domain that you are transferring.

4. Save and close `dbdomain.space_mapping.properties`.

5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - Windows: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
   - UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

**Parent topic:** Configuring WebSphere Portal to use DB2  
**Previous topic:** Creating remote databases  
**Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as **release.DbName**, **jcr.DbName**, **feedback.DbName**, and **likeminds.DbName**. For example:

  - `release.DbName` = `release`
  - `jcr.DbName` = `jcrdb`
  - `feedback.DbName` = `fdbkdb`
  - `likeminds.DbName` = `lmdb`
  - `community.DbName` = `commdb`
  - `customization.DbName` = `custdb`

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace `dbdomain` with:
  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`

- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
- dbdomain(DbType
- dbdomain(DbName
- dbdomain(DbUrl
- dbdomain(DbSchema

If you use the same values for all four properties across the release, customization, community, and JCR domains, the
database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not
the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In
other words, this value must be unique for the database domain.

1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out
the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each
instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your
environment.
   A. For dbdomain(DbType, type db2.
   B. For dbdomain(DbName, type the name of the WebSphere Portal domain database. Notes:
      - This value is also the database element in the dbdomain(DbUrl property.
      - This value is the TCP-IP alias for the database.
   C. For dbdomain(DbSchema, type the schema name of the database domain. Note: Review your target database
      management system documentation to define a valid schema name. Some database management systems have
      schema name restrictions that you need to understand.
   D. For dbdomain.DataSourceName, type the name of the data source that WebSphere Portal uses to communicate
      with its databases. Do not use the following reserved words:
         - releaseDS
         - communityDS
         - customizationDS
         - jcrDS
         - lmdbDS
         - feedback
   E. For dbdomain(DbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value
      must conform to the JDBC URL syntax specified by the database. Note: The database element of this value should
      match the value of DbName.
   F. For dbdomain(DbUser, type the user ID for the database administrator.
   G. For dbdomain(DbPassword, type the password for the database administrator.
   H. Optional: For dbdomain(DbRuntimeUser, type the user ID of the database user that should be used by WebSphere
      Portal to connect to the database at runtime.
   I. If dbdomain(DbRuntimeUser is specified, you must set dbdomain(DbRuntimePassword to be the password of the
      runtime database user.
   J. For dbdomain.DBA.DbUser, type the database administrator user ID for privileged access operations during creation
      of the database.
K. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.

L. For `dbdomain.XDbName`, type the database loop back alias that needs to be set if you plan to use the `create-database` task. **Note:** Required only for local databases using a JDBC Type 2 driver.

M. For `dbdomain.DbNode`, type the value for the node database. Set this value if you want to call `create-database`. **Note:** Required only for local databases.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For `source.domain.DbType`, type of the database you are currently configured to use. The value for `source.domain(DbType)` is Derby by default.

B. For `source.domain.DbName`, type the name of the database domain you are currently using.

C. For `source.domain.DbSchema`, type current schema identifier for objects within the database for this domain.

D. For `source.domain.DataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For `source.domain.DbUrl`, type the url currently used to access your database.

F. For `source.domain.DbUser`, type the name of the user accessing this database.

G. For `source.domain.DbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplcDbType.properties`.

   A. For `db2.DbDriver`, type the name of the JDBC driver class.

   B. For `db2.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For `db2.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.
Configuring JCR collation support

View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties

1. Stop the WebSphere Portal server:
   ```
   ./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password
   ```

2. If using a remote DB2 server, copy the following files from the WebSphere Portal server to a temporary directory on the remote DB2 server:
   ```
   PortalServer_root/jcr/prereq.jcr/config/collation.jar
   PortalServer_root/jcr/prereq.jcr/config/registerCollationUDFTemplate.sql
   ```

3. Set up collation on the database where the JCR domain is located.
   A. Change to the directory `db2_instance_owner_home/sqlib/function`.
   B. Execute the command: **Remote DB2**:
      ```
      db2_instance_owner_home/sqlib/java/jdk/bin/jar -xvf temporary location/collation.jar icm/CollationUDF.class
      ```
      **Local DB2**:
      ```
      db2_instance_owner_home/sqlib/java/jdk/bin/jar -xvf PortalServer_root/jcr/prereq.jcr/config/collation.jar icm/CollationUDF.class
      ```
   C. If using a local DB2, change to the directory `wp_profile_root/PortalServer/jcr/config`. If using a remote DB2, change to the temporary directory where you copied the files in a previous step.
   D. Open the file `registerCollationUDFTemplate.sql` and change all `SCHEMA` references to the JCR schema; for example, `JCR`. The value set for `SCHEMA` should match the value set for the `jcr.DbSchema` property. You specify `jcr.DbSchema` in the configuration file `wkplc_comp.properties` when you modify database properties.

4. Connect to the JCR database by running `db2` connect to `<jcrdb>` user `<userid>` using `<password>`. Execute the following command. **Remote DB2**:
   ```
   db2 -tvf temporary location/registerCollationUDFTemplate.sql
   ```
   **Local DB2**:
   ```
   db2 -tvf wp_profile_root/PortalServer/jcr/config/registerCollationUDFTemplate.sql
   ```

5. Edit the `icm.properties` file, located in `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm` directory. Add the following section to the end of the file:
   ```
   # Enable/Disable collation support for all DB2 platforms
   ```
# Disabled by default
jcr.query.collation.db2.enabled = true

# Database specific collation mappings
# These mappings apply map a Java locale name into a collation name
# supported by the underlying database.
# Example mappings for DB2 platform

# English
jcr.query.collation.en = en

# Swedish
jcr.query.collation.sv = sv

jcr.query.collation.zh = zh
jcr.query.collation.de = de
jcr.query.collation.da = da
jcr.query.collation.hu = hu
jcr.query.collation.jp = jp

6. Enter the following command to start the WebSphere Portal server:

```
./startServer.sh WebSphere_Portal
```
Configure WebSphere Portal to use DB2

View the steps to manually transfer data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
- Ensure that the following prerequisites are met:
  - Supported database software is installed.
  - Databases and users are set up.

**Prerequisites**
- Installing DB2
- Create users
- Creating remote databases
- Modifying database properties

**Tips:**
- To run these tasks as a non-root user, you must first run the task chown -R non-root_userWebSphereDir.
- If you are transferring from Oracle, the **open_cursors** setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

**Steps for transferring data to another supported database**

1. **Steps for transferring data to another supported database**
   - If you are running a type 2 connection, before transferring data edit the **db2cli.ini** file. Failure to follow these steps will cause the database transfer to hang at the task **action-process-constraints**.
     - **A.** Locate the file `/home/db2inst1/sqlib/cfg/db2cli.ini`.
     - **B.** Add the following to the end of the file. Leave an empty line after **ReturnAliases=0.[COMMON]**
       ```
       [COMMON]
       DYNAMIC=1
       ReturnAliases=0
       ```

2. **Steps for transferring data to another supported database**
   - Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.

3. **Steps for transferring data to another supported database**
   - Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2.
     ```
     /home/db2inst1/sqlib/db2profile
     where db2inst1 represents your database instance**Note:** You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.
     ```

4. **Steps for transferring data to another supported database**
   - Enter the following commands to validate configuration properties...
     ```
     ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
     ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
     ```

5. **Steps for transferring data to another supported database**
   - From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.

---

**Operating systems:** AIX, HP-UX, i5/OS, Linux, Solaris, Windows
6. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td>./stopServer.sh server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

7. Transfer the database:
   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.
   
   A. Change to the directory `wp_profile_root/ConfigEngine`.
   
   B. Enter the following command: 
   ```
   ./ConfigEngine.sh database-transfer -DWasPassword=password
   ```
   **Note:**
   - To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:
   ```
   ./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
   ```
   - If you have been storing data in IBM® Derby for a long time, database transfer could fail with OutOfMemory exceptions. If database transfer fails, add the following property to the command in this step: 
   ```
   ./ConfigEngine.sh database-transfer -DDbtJavaMaxMemory=1536M -DWasPassword=password
   ```

   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

8. After transferring the database tables, perform a `reorg` check to improve performance. Perform this step for each database alias in the property file.

   A. Connect to a database with the following command:
   ```
   db2 connect to database_alias user db2admin_userid using password
   ```
   **Note:** Additional options might be required if additional security has been installed. Refer to DB2 Universal Database commands by example for links to the command reference.

   B. After it is connected, run the following commands from the DB2 prompt:
   ```
   db2 reorgchk update statistics on table all > xyz.out
   ```
   Look in the reorg column for entries marked with a * (star or asterisk) in the file `xyz.out`. For each line with a *, note the `tablename` and run the following command for each `tablename`:
   ```
   db2 reorg table tablename db2 terminate db2rbind database_name -l db2rbind.out -u db2_admin -p password
   ```
   D. The output file `db2rbind.out` is only created when there is an error for the `db2rbind` command.

9. Change to the directory `wp_profile_root/bin`.

10. Enter the following command to start the WebSphere Portal server:
    ```
    ./startServer.sh WebSphere_Portal
    ```

**Parent topic:** Configuring WebSphere Portal to use DB2

**Previous topic:** Configuring JCR collation support

**Next topic:** Configuring DB2 for large file handling in Web Content Management
Configuring DB2 for large file handling in Web Content Management

If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties
  - Configure WebSphere Portal to use DB2

**Note:** You only need to perform these steps if you are using Web Content Management.

1. Log into the WebSphere Application Server administrative console.
2. Click **Resources > JDBC > Data sources**.
3. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
4. Select the name of the data source that is defined in `wkplc_comp.properties` for the JCR database domain. The default data source is `wpdbDS`.
5. Click **Custom properties**.
6. Ensure that the `fullyMaterializeLobData` property is set to `false`.

---

**Parent topic:** Configuring WebSphere Portal to use DB2

**Previous topic:** Configure WebSphere Portal to use DB2

**Next topic:** Changing driver types
Changing driver types

WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Before you begin, ensure that the following conditions are met:

- The WebSphere Portal database has been successfully transferred to DB2 using the database-transfer configuration task.
- The files wkplc_comp.properties and wkplc_dbtype.properties have been modified to set the correct values for the DB2 drivers that you are switching to:

  In the file wkplc_comp.properties set each `<Domain>.DbUrl` property using the following formats:
  # db2 (type 2):        { jdbc:db2:wpsdb }
  # db2 (type 4):        { jdbc:db2://<YourDatabaseServer>:50000/wpsdb:returnAlias=0; }

  In the file wkplc_dbtype.properties set the `db2.DbLibrary` property using the following format:
  # For DB2 Type 2 driver use `<SQLLIB>/java/db2java.zip`
  # For DB2 Type 4 driver use `<SQLLIB>/java/db2jcc.jar:<SQLLIB>/java/db2jcc_license_cu.jar`

  In the file wkplc_dbtype.properties set the `db2.DbDriver` property using the following format:
  # For DB2 Type 2 driver use COM.ibm.db2.jdbc.app.DB2Driver
  # For DB2 Type 4 driver use com.ibm.db2.jcc.DB2Driver

  **Note:** If WebSphere Portal is installed on the same machine as the DB2 server and you switch from a JDBC Type 4 connection to a JDBC Type 2 connection, verify that you have created the alias names for the DB2 databases as described in Creating remote databases and that the alias names are specified for the databases in the file wkplc_comp.properties.

  When switching from a JDBC Type 2 connection to a JDBC Type 4 connection, remove the database alias names and refer to the databases directly. This is required because of a limitation in the DB2 Universal JDBC driver.

- **Prerequisites**

  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties
  - Configure WebSphere Portal to use DB2
  - Configuring DB2 for large file handling in Web Content Management

1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.
2. Using a JDBC Type 2 driver only, export the DB2 user profile that you created when installing DB2 onto the administrative user using the following command. This command gives that administrative user rights over DB2. 

   ```
   /home/db2inst1/sqlib/db2profile
   ```

   where `db2inst1` represents your database instance

   **Note:** You must complete this step before running database tasks and before enabling security to create a suitable JDBC driver.

3. Enter the following commands to validate configuration properties:

   ```
   /ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ```
4. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td><code>./stopServer.sh server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td><code>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

6. Change to the directory `wp_profile_root/ConfigEngine`.

7. To change from one supported driver to the other, run the following task to connect the database, including only the domains that require the switch:

   ```bash
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
  
   ```

8. Change to the directory `wp_profile_root/bin`.

9. Enter the following command to start the WebSphere Portal server:

   ```bash
   ./startServer.sh WebSphere_Portal
   ```

Parent topic: Configuring WebSphere Portal to use DB2

Previous topic: Configuring DB2 for large file handling in Web Content Management
Configuring WebSphere Portal to use DB2 for z/OS

View information on installing and setting up DB2 for z/OS to work with WebSphere Portal.

1. **Installing DB2 for z/OS**
   - View information on how to install DB2 for z/OS for use with IBM® WebSphere® Portal.

2. **Creating users**
   - Complete this procedure to set up users for IBM DB2 Universal Database™ for z/OS® to work with IBM WebSphere Portal.

3. **Creating remote databases**
   - A remote database resides on a different machine than WebSphere Portal. You must manually create the required databases before configuring WebSphere Portal to work with DB2 for z/OS.

4. Optional: **Creating the Java Content Repository database**
   - View information on setting up your Java Content Repository database to work with WebSphere Portal.

5. Optional: **Assigning custom table spaces**
   - The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

6. **Modifying database properties**
   - This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

7. **Configuring WebSphere Portal to use DB2 for z/OS**
   - View information on manually transferring data to the DB2 for z/OS you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository to DB2 for z/OS. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

**Parent topic:** Configure WebSphere Portal to use a remote database
Installing DB2 for z/OS

View information on how to install DB2 for z/OS for use with IBM® WebSphere® Portal.

The following prerequisites must exist separately from the WebSphere Portal installation:

- If you are using the DB2 Legacy Type 2 JDBC driver, configure your DB2 Connect client with the following commands:
  - `db2 update dbm cfg using tp_mon_name WAS`
  - `db2 update dbm cfg using spm_name hostname`, where `hostname` is the host name for the machine where the DB2 Connect client is installed (same as portal machine).

- If you are using the DB2 Legacy Type 2 JDBC driver, enable extended shared memory usage with the following commands:
  - `export EXTSHM=ON`
  - `db2set DB2ENVLIST=EXTSHM`
  - `db2start`
  - To make the change permanent, you must add the environment variable by adding the following to the `profile.env` file:
    ```
    DB2ENVLIST='EXTSHM'
    ```
  - in `/home/db2inst/sqllib/userprofile add: export EXTSHM=ON`

  **Note:** The shell must be reopened before restarting DB2 for z/OS.

- Both type 2 and type 4 drivers are supported. If you are using the DB2 Legacy Type 2 JDBC driver, the DB2 Connect client must be installed on the same machine as WebSphere Portal to connect to the remote database.

- The DB2 subsystem must be on a supported z/OS platform.

- Update the BP8K0 catalog bufferpool to 35,000 buffers before performing database transfer. You should change this value based on your environment. The SYSIBM.SYSDATABASE table resides in this bufferpool and is used extensively by DB2 for z/OS during database transfer.

To use DB2 for z/OS as the database software for WebSphere Portal, you must have DB2 for z/OS installed on your z/OS system. Refer to the DB2 for z/OS product documentation for instructions. Refer to the Planning for DB2 for z/OS topic for additional considerations for configuring DB2 for z/OS as the WebSphere Portal database.

**Parent topic:** Configuring WebSphere Portal to use DB2 for z/OS

**Next topic:** Creating users
Creating users

Complete this procedure to set up users for IBM® DB2 Universal Database™ for z/OS® to work with IBM WebSphere® Portal.

Before you begin: You should have completed Installing DB2 for z/OS.

- **Prerequisites**
  - Installing DB2 for z/OS

- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

- If you are configuring multiple WebSphere Portal instances to use a single DB2 for z/OS subsystem, you must create a unique database user for each WebSphere Portal instance. By default all database table names include the name of the database user used to access the data. Therefore, to prevent table name conflicts, you must create a unique database user for each WebSphere Portal instance on the shared DB2 for z/OS subsystem.

- Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Use the following steps to grant access permissions to the database users. Repeat these steps for each WebSphere Portal instance you are setting up.

1. Create all database user IDs in the security product you are using on z/OS. For `jcrschema`, the database schema name for IBM Java Content Repository data, create a group and connect it to the database user ID for Java Content Repository data, `jcr`. The following sample shows the RACF definition of such a user ID and group, where `jcr` is the database user ID for Java Content Repository data, `yourDefaultUserGroup` is your default RACF group for database user IDs, `jcrschema` is the database schema name for Java Content Repository data, and `yourDefaultGroup` is your default RACF group for groups. If you have some other security product such as Top Secret or ACF2 instead of RACF, translate the sample RACF definition into the appropriate syntax before running the job.

   ```racf
   ADDUSER jcr DFLTGRP(yourDefaultUserGroup) NAME('WAS DB2 ACCESS USER')
   PW USER(jcr) NOINTERVAL
   ALU jcr PASSWORD(********) NOEXPIRED
   ADDGROUP jcrschema SUPGROUP(yourDefaultGroup)
   CONNECT jcr GROUP(jcrschema)
   ```

2. Run the following SQL statements using a tool like SPUFI while logged on to the DB2 subsystem to grant appropriate rights on the newly created databases. If you are configuring multiple WebSphere Portal instances to use a single DB2 for z/OS subsystem, be sure to use the database user associated with the database user ID you created for each WebSphere Portal instance you are setting up. (C) create/alter tablespaces
   (C) create/alter tables
   (C) create/alter indice;
   (C+R) read/write data
   (C) - at configuration time
   (R) - at runtime

Note: Make sure your vmmdb user follows these same standards.

WebSphere Portal instance you are setting up.
GRANT USE OF ALL BUFFERPOOLS TO releaseusr;
GRANT USE OF ALL BUFFERPOOLS TO communityusr;
GRANT USE OF ALL BUFFERPOOLS TO customizationusr;

GRANT DBADM ON DATABASE jcrdbnameonzos TO jcr;
GRANT USE OF ALL BUFFERPOOLS TO jcr;
GRANT DBADM ON DATABASE fdbkdbnameonzos TO feedback;
GRANT USE OF ALL BUFFERPOOLS TO feedback;
GRANT DBADM ON DATABASE lmdbnameonzos TO lmdbusr;
GRANT USE OF ALL BUFFERPOOLS TO lmdbusr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO releaseusr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO communityusr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO customizationusr;

GRANT SELECT ON SYSIBM.SYSTABLES TO releaseusr;
GRANT SELECT ON SYSIBM.SYSTABLES TO communityusr;
GRANT SELECT ON SYSIBM.SYSTABLES TO customizationusr;

GRANT SELECT ON SYSIBM.SYSCOLUMNS TO jcr;
GRANT SELECT ON SYSIBM.SYSTABLES TO jcr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO fdbkdbusr;
GRANT SELECT ON SYSIBM.SYSTABLES TO fdbkdbusr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO lmdbusr;
GRANT SELECT ON SYSIBM.SYSTABLES TO lmdbusr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO releaseusr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO communityusr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO customizationusr;

GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO jcr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO fdbkdbusr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO lmdbusr;
GRANT SELECT ON SYSIBM.SYSRELS TO releaseusr;
GRANT SELECT ON SYSIBM.SYSRELS TO communityusr;
GRANT SELECT ON SYSIBM.SYSRELS TO customizationusr;

GRANT SELECT ON SYSIBM.SYSRELS TO jcr;
GRANT SELECT ON SYSIBM.SYSRELS TO fdbkdbusr;
GRANT SELECT ON SYSIBM.SYSRELS TO lmdbusr;
GRANT USE OF STOGROUP jcrstogroup TO jcr;
GRANT CREATEIN, DROPIN ON SCHEMA jcrschema TO jcr;
GRANT SELECT ON SYSIBM.SYSTABLESPACE TO jcr;
GRANT SELECT ON SYSIBM.SYSVIEWS TO jcr;
GRANT SELECT ON SYSIBM.SYSDUMMY1 TO jcr;
GRANT SELECT ON SYSIBM.SYSTRIGGERS TO jcr;
GRANT SELECT ON SYSIBM.SYSINDEXPART TO jcr;
GRANT SELECT ON SYSIBM.SYSINDEXES TO jcr;
GRANT SELECT ON SYSIBM.SYSSYNONYMS TO jcr;

where:
- releasenameonzos, communitynameonzos, customizationnameonzos, and releaseusr, communityusr, customizationusr represent the databases and database users, respectively, of the WebSphere Portal instance you are setting up. (These users must be created on the host system.)

- jcrdbnameonzos and jcr are the database and database user, respectively, for Content Repository data.
- feedbackdbnameonzos and feedback are the database and database user, respectively, for Feedback data.
- lmdbnameonzos and lmdbusr are the database and database user, respectively, for Likeminds data.
- jcrschema is the database schema name for Content Repository data.

3. Grant the necessary access rights to all users who might require them. Depending on the architecture that you choose, these users might include Java Content Repository and Feedback users.

Parent topic: Configuring WebSphere Portal to use DB2 for z/OS
Previous topic: Installing DB2 for z/OS
Next topic: Creating remote databases
Creating remote databases

A remote database resides on a different machine than WebSphere Portal. You must manually create the required databases before configuring WebSphere Portal to work with DB2 for z/OS.

Before you begin, complete the following prerequisites:

- These instructions assume that you will use a local IBM® DB2 Universal Database™ Enterprise Server Edition Connect to connect to a remote IBM DB2 Universal Database for z/OS® server, and that IBM WebSphere® Application Server, WebSphere Portal, and DB2 Connect will be installed on the same machine.

- To run SQL statements, you can use a tool like SPUFI. The specified statements include CREATE STOGROUP, CREATE DATABASE and CREATE TABLESPACE. For CREATE STOGROUP you have to replace the icmvolumes and icmvcat variables with volume serial numbers and a catalog name for your environment. For CREATE TABLESPACE you can specify a specific BUFFERPOOL. Refer to the DB2 for z/OS SQL Reference for a description of these options.

- Ensure that a TEMP database is created for your subsystem. You can use the following statements to create a TEMP database:
  
  ```sql
  CREATE DATABASE db_name AS TEMP;
  CREATE TABLESPACE ts_name IN db_name;
  ```

- Log on to the DB2 subsystem on the host server. DB2 system administrator rights are needed to create the databases.

- Replace variables as follows:

  - releasenameonzos, communitynameonzos, and customizationnameonzos are the WebSphere Portal databases for WebSphere Portal and Member Manager data.
  - fdbkdbnameonzos and fdbkdbts are the database and table space, respectively, for Feedback data.
  - lmdbnameonzos and lmdbs are the database and table space, respectively, for LikeMinds data.

- Because large objects are stored in columns that can become very large, logging changes to these columns requires a huge amount of log space. For this reason, large object (LOB) logging is disabled by default for tablespaces that contain such data. With LOB logging disabled, you can recover full backups, but not incremental backups that can be used for point in time recovery. To recover point in time backups, you must enable LOB logging. For detailed instructions, see Technote 1306637, Managing LOB logging in DB2 for z/OS.

- **Prerequisites**
  - Installing DB2 for z/OS
  - Creating users

Use the following steps to set up WebSphere Portal with a remote instance of DB2 for z/OS. Refer to Planning for DB2 for z/OS for a list of databases and table space names. If you are configuring multiple WebSphere Portal instances to use a single DB2 subsystem, be sure to use the database and table space names associated with the WebSphere Portal instance you are setting up.

1. ```sql
   CREATE DATABASE releasenameonzos CCSID UNICODE;
   ```

2. ```sql
   CREATE DATABASE communitynameonzos CCSID UNICODE;
   ```

3. ```sql
   CREATE DATABASE customizationnameonzos CCSID UNICODE;
   ```

4. Execute the steps in the topic Creating the Java Content Repository database.

5. ```sql
   CREATE DATABASE fdbkdbnameonzos CCSID UNICODE;
   CREATE TABLESPACE fdbkdbts IN fdbkdbnameonzos USING STOGROUP SYSDEFLT PRIQTY 5000 SECQTY 500 LOCKSIZE ROW;
   ```

6. ```sql
   CREATE DATABASE lmdbnameonzos CCSID UNICODE;
   ```
8. CREATE TABLESPACE lmdbts IN lmdbnameonzos USING STOGROUP SYSDEFLT PRIQTY 5000 SECQTY 500;
Creating the Java Content Repository database

View information on setting up your Java Content Repository database to work with WebSphere Portal.

First, you need to create multiple databases for scalability.

The IBM Java Content Repository database grows in size as the IBM® WebSphere® Portal server is used. To support this growth, which includes the dynamic creation of many new tables within the database, the WebSphere Portal server allows you to create multiple databases within a single IBM DB2 Universal Database™ for z/OS® subsystem. A minimum of two databases is recommended, but more can be created if necessary. A single database setup is not recommended; it will quickly become constrained. See the Performance guides on the product documentation Web site for more information.

Each database that is created must begin with a common prefix; there is no convention required for the remainder of the name. For example, to create \( xx \) number of databases, you may choose to use the following commands:

\[
\text{CREATE DATABASE JCRDB01}
\]

\[
\text{CREATE DATABASE JCRDB02}
\]

\[
\ldots
\]

\[
\text{CREATE DATABASE JCRDBxx}
\]

In this case, \( JCR \) is the common prefix. You also have to select a maximum number of user-defined tables (UDT) that each database will be allowed to hold; the recommended value is 400.

- **Prerequisites**
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases

A sample script of the commands that you can use to create the Java Content Repository database in your DB2 for z/OS installation is shown below. The sample demonstrates the creation of a single database. To follow the recommendation of creating at least two databases, duplicate the commands for each additional database as indicated below. Find a template of these commands in the file `PortalServer_root installer/wp.config/config/templates/db/db2_zos/jcr_sample.sql`.

**Notes:**
- It is not necessary to duplicate every tablespace definition in every database.
- In order to run the commands shown in the sample, you must know the database name and the IDs of the MVS volumes where the Java Content Repository database tables will be stored.
- Edit the template file for your installation environment before running the commands shown in the sample:
  - Replace `jcrdbnameX` with the name of the database. Remember that each database name must begin with a common prefix.
  - Replace `stogroup` with the name of your storage group.
  - Replace `icmvolumes` with the IDs of the MVS volumes where the Java Content Repository database tables will be stored. These values are assigned by the database administrator.
  - Replace `icmvcat` with the name of the virtual catalog.
  - Replace `jcr` with the name of database user ID.
  - Replace `4kbp` with the name of your 4K bufferpool.
  - Replace `32kbp` with the name of your 32K bufferpool.
  - Replace `jcrschema` with the schema name of your Java Content Repository domain.
--DROP DATABASE jcdbNameX
--DROP STOGROUP stogroup
--DROP ALIAS jcrschema.ICMFICTITIOUS

CREATE STOGROUP stogroup VOLUMES (icmvolumes) VCAT icmvcat
GRANT USE OF STOGROUP stogroup to jcr

CREATE DATABASE jcdbNameX STOGROUP stogroup BUFFERPOOL 4kbp INDEXBP 4kbp CCSID UNICODE

**Note:** The following two tablespace commands must be created in every database:

CREATE TABLESPACE ICMLFQ32 IN jcdbNameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICMTSQ04 IN jcdbNameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

**Note:** The following tablespaces are required in the first database only:

CREATE TABLESPACE ICMLFQ04 IN jcdbNameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICMTSQ04 IN jcdbNameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSTADO IN jcdbNameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRIDLSTS IN jcdbNameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRSCLSTS IN jcdbNameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRISLSTS IN jcdbNameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRGCLSTS IN jcdbNameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE PRIGLSTS IN jcdbNameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSTDPV IN jcdbNameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCCLSTS IN jcdbNameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICSDOAC IN jcdbNameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RESMLSTS IN jcdbNameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COLNLSTS IN jcdbNameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USERLSTS IN jcdbNameX USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE USEGLSTS IN jcdbNameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ACCLLSTS IN jcdbNameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE SYSCLSTS IN jcdbNameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

1831
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CACLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CPERLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATDDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITTRLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITALLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COVDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CPERLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COVDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NLSKLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE MAXKLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NLSLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE CACLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITALLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COMFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COMALSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE COMDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITVILSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITVDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ITTDLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NLSKLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE MAXKLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NLSLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ATTGLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE NLSKLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE ICUT301 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TIELLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE REPLLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RI11LSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE XDOFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE XDOOLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE IDELLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TEICLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE SYAELSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TS00201 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TS00202 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TS00203 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TS00204 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TS00205 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TS00206 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TS00207 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TS00208 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICMACTLS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICMACLS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT301 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT311 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT321 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT331 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT341 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TSPEJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TSTAJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RIDSJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE WSNOJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE LOB TABLESPACE ICMLOBT1 IN jcrdbnameX LOG NO
CREATE LOB TABLESPACE ICMLOBT2 IN jcrdbnameX LOG NO
CREATE LOB TABLESPACE ICMLOBT3 IN jcrdbnameX LOG NO
CREATE ALIAS jcrschema.ICMFICTITIOUS FOR SYSIBM.SYSDUMMY1
CREATE TABLESPACE DLKRJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE LKRLJRTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.
- When using IBM® DB2 Universal Database™ for z/OS® as a data store, WebSphere Portal requires that its indexes are not padded. Therefore, you must set the DSNZPARM parameters to `RETVLCFK=NO` or `PADIX=NO`, or both.

**Prerequisites**
- Installing DB2 for z/OS
- Creating users
- Creating remote databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`
   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
     - release
     - community
     - customization
     - jcr
     - feedback
     - likeminds
3. Assign a table space to each `.tablespace` entry in the mapping file. Assignments to `.indexspace` entries are ignored. The table space name must be qualified by the database name and prepended by the keyword `IN` and a space. For example: `community.COMP_INST.tablespace=IN COMM8KSPACE`
   Repeat this step for each domain that you are transferring.
4. Save and close `dbdomain.space_mapping.properties`.
5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - Windows: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
- UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

**Parent topic:** Configuring WebSphere Portal to use DB2 for z/OS

**Previous topic:** Creating the Java Content Repository database

**Next topic:** Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- Prerequisites
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

Tip: For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Working with properties files:
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:
  - release.DbName=release
  - jcr.DbName=jcrdb
  - feedback.DbName=fdbkdb
  - likeminds.DbName=lmdb
  - community.DbName=commdb
  - customization.DbName=custdb
- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If `DbUser`, `DbUrl`, and `DbPassword` are not the same across domains, the value for `DataSourceName` must differ from the `DataSourceName` of the other domains. In other words, this value must be unique for the database domain.

**Note:** To successfully transfer data from the JCR domain, you must use the DDF location value for the value of `jcr.DbName` field when setting up IBM DB2 Universal Database™ for z/OS®. You can locate the name of the DDF location value in the IBM DB2 Universal Database for z/OS sdsnsamp dataset, member DSNTIJUZ, or by running the following DB2 command:
```
db2 subsystem prefix display ddf
```

1. Locate the following files and create a backup copy of each before changing any values:
   - `wp_profile_root/ConfigEngine/properties/wkplc.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_comp.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties`

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In `wkplc_comp.properties`, most properties are repeated for each domain.

2. Use a text editor to open the properties file `wkplc_comp.properties` and modify the values to correspond to your environment.
   
   A. For `dbdomainDbType`, type `db2_zos`.
   
   B. For `dbdomainDbName`, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the `dbdomainDbUrl` property.
   
   C. For `dbdomainDbSchema`, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   
   D. For `dbdomainDbNameOnZos`, type the name of the WebSphere Portal database on DB2 for z/OS. **Note:**
      - If running DB2 for z/OS as a remote database, set the value to the name of the remote database for the domain.
      - If WebSphere Portal is running on z/OS with DB2 for z/OS, set the value equal to the value of `DbName`.
   
   E. For `dbdomainDataSourceName`, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - `releaseDS`
      - `communityDS`
      - `customizationDS`
      - `jcrDS`
      - `lmdbDS`
      - `feedback`
   
   F. For `dbdomainDbUrl`, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:** The database element of this value should match the value of `DbName`.
   
   G. For `dbdomainDbUser`, type the user ID for the database administrator.
   
   H. For `dbdomainDbPassword`, type the password for the database administrator.
I. For `dbdomain(DbTablespace`, type the name of the DB2 for z/OS tablespace.
J. For `dbdomain(DbStorageGroup`, type the name of the storage group for the database.
K. For `dbdomain(DbVolumes`, type the volumes for the database.
L. For `dbdomain(DbVcat`, type the VCAT for the database.
M. For `dbdomain(Db4KBufferPoolName`, type the 4K bufferpool name for the database.
N. For `dbdomain(Db32KBufferPoolName`, type the 32K bufferpool name for the database.
O. For `dbdomain(DbIndex4KBufferPoolName`, type the 4K bufferpool name for the database. If you choose to use the default bufferpool value BP3, verify that this bufferpool is active.
P. For `dbdomain(TablespaceTrackMod`, set the value to determine TRACKMOD attribute of all tablespaces to use the specified value. Refer to the DB2 for z/OS documentation before changing this value.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.
A. For `source.domainDbType`, type of the database you are currently configured to use. The value for `source.domainDbType` is Derby by default.
B. For `source.domainDbName`, type the name of the database domain you are currently using.
C. For `source.domainDbSchema`, type current schema identifier for objects within the database for this domain.
D. For `source.domainDataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.
E. For `source.domainDbUrl`, type the url currently used to access your database.
F. For `source.domainDbUser`, type the name of the user accessing this database.
G. For `source.domainDbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkpic_dbtype.properties`.
A. For `db2_zos.DbDriver`, type the name of the class that SqlProcessor uses to import SQL files.
B. For `db2_zos.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.
C. For `db2_zos.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.
D. For `db2_zos.DbDriverType`, type the number of the driver type for the database.
E. For `db2_zos.DbLocationName`, type the DB2 location name. This value is set in the installation job DSNTIJUZ.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.
A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Configuring WebSphere Portal to use DB2 for z/OS

**Previous topic:** Assigning custom table spaces

**Next topic:** Configuring WebSphere Portal to use DB2 for z/OS
Configuring WebSphere Portal to use DB2 for z/OS

View information on manually transferring data to the DB2 for z/OS you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository to DB2 for z/OS. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

- Prerequisites
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases
  - Modifying database properties

Steps for transferring data to another supported database

1. If you are running a type 2 connection, before transferring data edit the db2cli.ini file. Failure to follow these steps will cause the database transfer to hang at the task action-process-constraints.
   
   A. Locate the file /home/db2inst1/sqllib/cfg/db2cli.ini.
   
   B. Add the following to the end of the file. Leave an empty line after ReturnAliases=0.[COMMON]
      
      DYNAMIC=1
      ReturnAliases=0

2. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.

3. Enter the following commands to validate configuration properties.
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password

4. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td>./stopServer.sh server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td>./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

6. Transfer the database:
   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.
   
   A. Change to the directory wp_profile_root/ConfigEngine.
   
   B. Enter the following command:
      ./ConfigEngine.sh database-transfer -DWasPassword=password

      **Note:** To select specific database domains to transfer, modify the -DTransferDomainList specified in the command
to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter
the following command: 

```
./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
```

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and
`wkplc_dbtype.properties` files and then repeat this step.

7. From the command line processor, reset the check pending status on the WebSphere Portal table spaces listed here.

CHECK DATA is a DB2 batch utility that is invoked through JCL or through the DB2 utility panels. Type the following
utility commands to check pending status:

```
check data tablespace releasenameonzos.TS280A
check data tablespace releasenameonzos.TS300A
check data tablespace releasenameonzos.TS2110A
check data tablespace releasenameonzos.TS830A
check data tablespace communitynameonzos.TS8000B
check data tablespace communitynameonzos.TS8011B
check data tablespace communitynameonzos.TS280B
check data tablespace customizationnameonzos.TS2110C
check data tablespace jcrdbnameonzos.ICMSFQ04
check data tablespace jcrdbnameonzos.TS2110D
```

where `releasenameonzos`, `communitynameonzos`, and `customizationnameonzos` are the names of your WebSphere
Portal databases, and `jcrdbnameonzos` is the name of your JCR database. Refer to the *Utility Guide and Reference for
DB2 for z/OS* for additional details.

8. Run the `RUNSTATS` utility as shown in the following example:

```
LISTDEF JCRDB2OS INCLUDE TABLESPACE JCRDB2OS.* BASE
RUNSTATS TABLESPACE LIST JCRDB2OS INDEX(ALL) KEYCARD TABLE(ALL)
LISTDEF JCRDB01 INCLUDE TABLESPACE JCRDB01.* BASE
RUNSTATS TABLESPACE LIST JCRDB01 INDEX(ALL) KEYCARD TABLE(ALL)
...```

9. Start the WebSphere Application Server.

10. Verify that the WebSphere Portal application server is running by opening the following URL in a browser:

```
http://hostname.companyname.com:port_number/wps/portal
```

where `hostname.companyname.com` is the fully qualified host name of the machine where WebSphere Portal is running and `port_number` is the transport port that is created by
WebSphere Application Server.

If you have performed database transfer in a clustered environment, you must explicitly copy the `wp_profile_root
/PortalServer/jcr/lib/com/ibm/icm/icm.properties` file from the primary node on which the `database-transfer` task
was processed to all secondary nodes. This ensures that the secondary nodes have the new JCR database values
specified in the `icm.properties` file.

1. Stop the portal server on the secondary node.
2. Copy the `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm/icm.properties` file from the primary node and
   replace the `icm.properties` file on the secondary node with the new file from the primary node.
3. Start the portal server on the secondary node.

Parent topic: Configuring WebSphere Portal to use DB2 for z/OS

Previous topic: Modifying database properties
Configuring WebSphere Portal to use Oracle

View information on installing and setting up Oracle to work with WebSphere Portal.

1. Installing Oracle
   View information on how to install Oracle for use with WebSphere Portal.

2. Creating users
   View the steps to set up users for Oracle to work with WebSphere Portal.

3. Creating databases
   View some important considerations before setting up Oracle databases to work with WebSphere Portal.

4. Modifying database properties
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and
   wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create
   databases, create users, or transfer data.

5. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an
   alternative to automatically setting up the database, you can manually create users and grant privileges.

6. Creating JCR table spaces
   This topic provides manual instructions for creating JCR table spaces for Oracle.

7. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When
   using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer
   to the target database system.

8. Configure WebSphere Portal to use Oracle
   This section provides information on how to manually transfer data to the Oracle database you have installed and set
   up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an
   alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete
   the database transfer task.

Parent topic: Configure WebSphere Portal to use a remote database
Installing Oracle

View information on how to install Oracle for use with WebSphere Portal.

You can use Oracle as the database software. You must install Oracle separately and before performing a database transfer.

Refer to Oracle product documentation for installation instructions.

Parent topic: Configuring WebSphere Portal to use Oracle
Next topic: Creating users
Creating users

View the steps to set up users for Oracle to work with WebSphere Portal.

Before you begin:

- You should have installed Oracle.

- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

**Prerequisites**

- Installing Oracle

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Start the Oracle SQL command line processor 'SQL*Plus' by entering `sqlplus /nolog` on the operating system command prompt

2. Log in to the Oracle database by executing command `connect user/password@database name`. For example: `system/manager@wpsdb` will log the administrative user system with a password of manager into the wpsdb database.

3. Create the WebSphere Portal user `dbdomain.DbUser`, where `dbdomain` is replaced by release, community, or customization.

   A. Log in to the database in which you want to create the new users.

   B. Create the users that you need in the appropriate databases by entering the following commands in the SQL*Plus tool, where `user_tablespace` is the default tablespace that is identified by the database administrator to store user objects and `temp_tablespace` is identified to store temporary objects. All names must be in lower case.

   ```sql
   SQL> create user releaseusr identified by password
   default tablespace user_tablespace
   temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO releaseusr;
   SQL> create user communityusr identified by password
   default tablespace user_tablespace temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO communityusr;
   SQL> create user customizationusr identified by password
   default tablespace user_tablespace temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO customizationusr;
   SQL> create user jcr identified by password
   default tablespace user_tablespace temporary tablespace temp_tablespace;
   GRANT UNLIMITED TABLESPACE TO jcr
   ```

4. Connect to the Feedback database:

   A. Enter the following command: `SQL> connect`
B. Enter user-name: username/password@dbname where username is an existing administrative user in the database. For example, system/manager@fdbkdb will log the administrative user system with a password of manager into the fdbkdb database.

C. Create the Feedback user:

```
SQL> create user feedback identified by password
    
default tablespace user_tablespace
    
temporary tablespace temp_tablespace;
    
GRANT UNLIMITED TABLESPACE TO feedback
```

D. Log out of the command line tool using the command SQL> exit.

5. Connect to the LikeMinds database:

A. Enter the following command:

```
SQL> connect
```

B. Enter user-name: username/password@dbname, where username is an existing administrative user in the database.

For example, system/manager@lmdb will log the administrative user system with a password of manager into the lmdb database.

C. Create the LikeMinds user:

```
SQL> create user lmdbusr identified by password
    
default tablespace user_tablespace
    
temporary tablespace temp_tablespace;
    
GRANT UNLIMITED TABLESPACE TO lmdbusr
```

D. Log out of the command line tool using the command SQL> exit.

6. Perform the following steps to verify that you can connect to the database with the user ID and password that you just created:

A. Enter the SQL> connect command to connect to the content database.

B. Enter user-name: username/password@dbname, where username is an existing administrative user in the database. For example, system/manager@jcrdb will log the administrative user system with a password of manager into the Oracle database.

Parent topic: Configuring WebSphere Portal to use Oracle

Previous topic: Installing Oracle

Next topic: Creating databases
Creating databases

View some important considerations before setting up Oracle databases to work with WebSphere Portal. For information about creating databases, refer to the Oracle product documentation. For information on the recommended database architecture and the databases you will need to create, see the Planning for Oracle topic. Be sure that all databases to be used with WebSphere Portal are created as UNİCODE character set databases.

If you are using remote Oracle 9 or 10 databases, you must also copy the ojdbc14.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM® WebSphere® Application Server Version 7.0, you must also copy the ojdbc6.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM WebSphere Application Server Version 6.1, you must also copy the ojdbc5.jar file from the remote Oracle server to the WebSphere Portal machine. The typical location is the oracle_home/sqldeveloper/jdbc/lib directory. Record the copy location on your local machine for future reference.

When creating Oracle databases for use with WebSphere Portal, you should consider the following information:
- The Oracle databases must be created manually before configuring WebSphere Portal.
- All databases must be created using UNİCODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle server will be remote or local to the WebSphere Portal installation.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle database in order for WebSphere Portal to communicate with the Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes:

  - db_block_size = 8192 bytes
  - db_cache_size = 307,200 bytes
  - db_files = 1024 files
  - log_buffer = 65536 bytes
  - open_cursors = 1500 cursors
  - pga_aggregate_target = 204,800 bytes
  - pre_page_sga = true
  - processes = 300 processes
  - shared_pool_size = 204,800 bytes

**Note:** If you are using IBM Java Content Repository, the open_cursors value may need to be increased based on the table count in the Java Content Repository schema.

- Raise the number of parallel servers as appropriate. For example, if you have more than 875 parallel servers, you should set the parallel_max_servers to 1200.
- The Oracle parameter CURSOR_SHARING allows similar SQL Statements to be shared when possible, which prevents parsing and establishing a new execution plan. The execution plan is used by Oracle to gather the data needed to satisfy a request. There are two options for CURSOR_SHARING, which are as follows:

  - **FORCE**
    - When you select this option, Oracle uses the same execution plan for all SQLs that are similar in value even if the values are different. When you use this option, the execution plan may not provide optimum performance. For example, similar SQLs with different values may behave differently when executed running the same plan.

  - **EXACT**
When you select this option, Oracle only shares the same execution plan for SQLs that are identical and use the same values. This option removes the risk of a SQL statement being executed when optimum performance conditions do not exist.

WebSphere Portal supports both options. Regardless of the option selected, portlet applications should not be affected. Contact your database administrator for further assistance on these options.

- **Prerequisites**
  - Installing Oracle
  - Creating users

**Parent topic:** Configuring WebSphere Portal to use Oracle  
**Previous topic:** Creating users  
**Next topic:** Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Installing Oracle
  - Creating users
  - Creating databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:

  - `release.DbName` = `release`
  - `jcr.DbName` = `jcrdb`
  - `feedback.DbName` = `fdbkdb`
  - `likeminds.DbName` = `lmdb`
  - `community.DbName` = `commdb`
  - `customization.DbName` = `custdb`

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace `dbdomain` with:
  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`

- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.

   A. For dbdomain.DbType, type oracle.
   B. For dbdomain.DbName, type the name of the WebSphere Portal domain database. \textbf{Note:} This value is also the database element in the dbdomain.DbUrl property.
   C. For dbdomain.DbSchema, type the schema name of the database domain. \textbf{Note:} Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The dbdomain.DbName should be the same value used for the dbdomain.DbSchema
      \hspace{1cm} \textbf{Restriction:} The value for dbdomain.DbSchema must equal the value for dbdomain.DbUser.
   D. For dbdomain.DataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For dbdomain.DbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. \textbf{Note:}
      - The database element of this value should match the value of DbName.
      - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following:
        jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME. \textbf{When database transfer is completed,} the WebSphere Portal server will be configured to use this single database server.
F. For `dbdomain.DbUser`, type the user ID for the database administrator. **Restriction:** The value for `dbdomain.DbUser` must equal the value for `dbdomain.DbSchema`.

G. For `dbdomain.DbPassword`, type the password for the database administrator.

H. For `dbdomain.DBA.DbUser`, type the database administrator user ID for privileged access operations during creation of the database.

I. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.

J. For `dbdomain.DbHome`, type the root location for the database. **Note:** This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For `source.domain.DbType`, type of the database you are currently configured to use. The value for `source.domain.DbType` is Derby by default.

B. For `source.domain.DbName`, type the name of the database domain you are currently using.

C. For `source.domain.DbSchema`, type current schema identifier for objects within the database for this domain.

D. For `source.domain.DataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For `source.domain.DbUrl`, type the url currently used to access your database.

F. For `source.domain.DbUser`, type the name of the user accessing this database.

G. For `source.domain.DbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

A. For `oracle.DbDriver`, type the name of the class that SqlProcessor uses to import SQL files.

B. For `oracle.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

C. For `oracle.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Configuring WebSphere Portal to use Oracle  
**Previous topic:** Creating databases  
**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

**Prerequisites**
- Installing Oracle
- Creating users
- Creating databases
- Modifying database properties

1. On the database server, make sure that the folder `db_1` contains the subfolders `your_oracle_instance/data` and `your_oracle_instance/index`. If this folder hierarchy does not exist, create it manually before you run the setup-database task. The setup-database task requires these folders to create database users. If these folders do not exist, the setup-database task will fail.

2. Change to the directory `wp_profile_root/ConfigEngine`.

3. To create the database users, type the following command:

   ```bash
   ./ConfigEngine.sh setup-database -DWasPassword=password
   
   Note: This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:
   
   `<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql`.
   ```

**Parent topic:** Configuring WebSphere Portal to use Oracle

**Previous topic:** Modifying database properties

**Next topic:** Creating JCR table spaces

**Related tasks**
Manually creating users and granting privileges for Oracle
Creating JCR table spaces
This topic provides manual instructions for creating JCR table spaces for Oracle.

- Prerequisites
  - Installing Oracle
  - Creating users
  - Creating databases
  - Modifying database properties
  - Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.
2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:
   - `&jcrdb` is the name of the database you created to store user data.
   - `&dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.
   
   Ensure that the ' ' is included in the variables when you substitute the values of your environment with these variables.
   
   Important: You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

   ```sql
   create tablespace ICMLFQ32 datafile '&dbpath./&jcrdb./data/'&jcrdb._ICMLFQ32_01.dbf' size 300M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLNF32 datafile '&dbpath./&jcrdb./data/'&jcrdb._ICMLNF32_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMVFQ04 datafile '&dbpath./&jcrdb./data/'&jcrdb._ICMVFPQ04_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMSFQ04 datafile '&dbpath./&jcrdb./data/'&jcrdb._ICMSFPQ04_01.dbf' size 150M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLSNDX datafile '&dbpath./&jcrdb./index/'&jcrdb._ICMLSNDX_01.dbf' size 10M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   ```

   A. Set the `size`, `autoextend`, and `maxsize` values according to your environment. For example, you may want to change the `maxsize` to a set value rather than `UNLIMITED`.
   B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.
   C. Refer to the Oracle command reference for more information about using the `create tablespace` command.

Parent topic: Configuring WebSphere Portal to use Oracle
Previous topic: Setting up databases
Next topic: Assigning custom table spaces
Assigning custom table spaces
The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the wp_profile_root/PortalServer/config/tablespsaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

**- Prerequisites**
- Installing Oracle
- Creating users
- Creating databases
- Modifying database properties
- Setting up databases
- Creating JCR table spaces

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file wp_profile_root/PortalServer/config/tablespsaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.index_name.indexspace
   For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword TABLESPACE and a space. For example: community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE
   Repeat this step for each domain that you are transferring.
4. Save and close dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
- UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

**Parent topic:** Configuring WebSphere Portal to use Oracle  
**Previous topic:** Creating JCR table spaces  
**Next topic:** Configure WebSphere Portal to use Oracle
Configure WebSphere Portal to use Oracle

This section provides information on how to manually transfer data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- Prerequisites
  - Installing Oracle
  - Creating users
  - Creating databases
  - Modifying database properties
  - Setting up databases
  - Creating JCR table spaces

Tips:

- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

Steps for transferring data to another supported database

1. Open a command prompt and change to the directory wp_profile_root/ConfigEngine.
2. Enter the following commands to validate configuration properties:
   ```
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```
3. From the same command prompt as the previous steps, change to the directory wp_profile_root/bin.
4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td>./stopServer.sh server1 -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>
5. Transfer the database. **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory `wp_profile_root/ConfigEngine`.

   B. Enter the following command: `./ConfigEngine.sh database-transfer -DWasPassword=password`

   **Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command: `./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password`

   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files. If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example: `SQL> execute dbms_stats.gather_schema_stats(ownname=>'jcr', cascade=> TRUE);`

7. Change to the directory `wp_profile_root/bin`.

8. Enter the following command to start the WebSphere Portal server: `./startServer.sh WebSphere_Portal`
Configuring WebSphere Portal to use Oracle RAC

View information on installing and setting up Oracle RAC to work with WebSphere Portal.

1. Installing Oracle RAC
   View information on how to install Oracle RAC for use with WebSphere Portal.

2. Create users
   View the steps to set up users for Oracle RAC to work with WebSphere Portal.

3. Creating databases
   View some important considerations before setting up Oracle RAC databases to work with WebSphere Portal.

4. Modifying database properties
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

6. Creating JCR table spaces
   This topic provides manual instructions for creating JCR table spaces for Oracle.

7. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

8. Configuring WebSphere Portal to use Oracle RAC
   This section provides information on how to manually transfer data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configure WebSphere Portal to use a remote database
Installing Oracle RAC

View information on how to install Oracle RAC for use with WebSphere Portal.

Before you begin:
- You should have completed reviewing the Planning for Oracle RAC section.
- Ensure the Oracle CRS (Cluster Ready Service) and Oracle RAC databases have been installed and configured on the primary and secondary nodes.
- Run the following commands on both RAC nodes to start global services daemon_(GSD), oracle listeners, and agents.
  - $ gsdctl start
  - $ lsnrctl start
  - $ agentctl start

You can use Oracle RAC as the database software for WebSphere Portal. You must install Oracle RAC separately and before performing WebSphere Portal database transfer.

Refer to Oracle product documentation for installation instructions.

Parent topic: Configuring WebSphere Portal to use Oracle RAC

Next topic: Create users
Create users

View the steps to set up users for Oracle RAC to work with WebSphere Portal.

Before you begin:
- You should have completed installing Oracle RAC.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

**Prerequisites**
- Installing Oracle RAC

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Create the users in the WebSphere Portal databases
   A. Log in to the database in which you want to create the new users.
   B. Create the users that you need in the appropriate databases by entering the following command in the SQL*Plus tool, where `user_tablespace` is the default tablespace that is identified by the database administrator to store user objects and `temp_tablespace` is identified to store temporary objects:

   ```sql
   SQL> create user username identified by password
   default tablespace user_tablespace
   temporary tablespace temp_tablespace;
   ```

   **Tip:** Balance the storage of user objects among tablespaces to prevent running out of space with overuse of `user_tablespace`. Also consider increasing the size of `user_tablespace` when handling a high volume of users.

2. Log in by entering the command `$ sqlplus` in SQL*Plus:

3. Enter the command `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example: `system/manager@wpsdb` will log the administrative user `system` with a password of `manager` into the wpsdb database.

4. Create the WebSphere Portal user `dbdomain.DbUser`, where `dbdomain` is replaced by release, community, customization, or JCR. All names must be in lower case. The following code uses the default user tablespace and the default temporary tablespace. Explicitly specify these tablespaces if you do not want to use the defaults.

   ```sql
   SQL> create user releaseusr identified by password;
   SQL> create user communityusr identified by password;
   SQL> create user customizationusr identified by password;
   SQL> create user jcr identified by password;
   ```

5. Connect to the content database by entering the command `SQL> connect`.

6. Enter the following, where `username` is an existing administrative user in the database. `user-name: username/password@dbname`. For example, `system/manager@jcrdb` will log the administrative user `system` with a password of `manager` into the Oracle RAC database.

7. Create the Oracle RAC users by entering the following: When creating the `jcr` user, grant all necessary privileges. If you do not grant privileges, you will receive the error `ICMADMIN lacks CREATE SESSION Privilege logon denied`
when you try to connect with the `jcruser`:

```
SQL> create user jcr identified by password default tablespace users temporary tablespace temp;
```

8. Connect to the Feedback database:

   A. Enter the following command: `SQL> connect`

   B. Enter `user-name: username/password@dbname` where `username` is an existing administrative user in the database.

      For example, `system/manager@fdbkdb` will log the administrative user system with a password of manager into the fdbkdb database.

9. Create the Feedback user:

    `SQL> create user feedback identified by password default tablespace users temporary tablespace temp;`

10. Connect to the LikeMinds database:

    `SQL> connect`

11. Enter `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example, `system/manager@lmdb` will log the administrative user system with a password of manager into the lmdb database.

12. Create the LikeMinds user:

    `SQL> create user lmdbusr identified by password default tablespace users temporary tablespace temp;`

13. Log out of the command line tool using the command `SQL> exit`.

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC

**Previous topic:** Installing Oracle RAC

**Next topic:** Creating databases
Creating databases

View some important considerations before setting up Oracle RAC databases to work with WebSphere Portal. When creating Oracle RAC databases for use with WebSphere Portal, you should consider the following information:

- The Oracle RAC databases must be created manually before configuring WebSphere Portal.
- All databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle RAC server will be remote or local to the WebSphere Portal installation.
- If using an earlier version of Oracle RAC (9i or earlier), ensure that Oracle RAC JVM is also installed.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle RAC database in order for WebSphere Portal to communicate with the IBM Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes are:
  
  - `db_block_size` = 8192 bytes
  - `db_cache_size` = 314,572,800 bytes
  - `db_files` = 1024 files
  - `log_buffer` = 65536 bytes
  - `open_cursors` = 1500 cursors
  - `pga_aggregate_target` = 209,715,200 bytes
  - `pre_page_sga` = true
  - `processes` = 300 processes
  - `shared_pool_size` = 209,715,200 bytes

  **Note:** If you are using IBM Java Content Repository, the `open_cursors` value may need to be increased based on the table count in the IBM Java Content Repository schema.

- **Prerequisites**
  - Installing Oracle RAC
  - Create users

Refer to the following instructions to create tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.

2. Create the tablespaces. **Important:** You must use the same tablespace names listed in the commands. The tablespace names cannot be customized or modified.

   A. Find and edit the SQL script `jcr_ora_tablespaces.sql` in the directory `wp_profile_root/ConfigEngine/work/db/oracle`.

   B. In the define section, replace the following variables with the values from your environment:
      - `jcrdb`
        - Name of the database you created to store user data.
      - `logfile`
        - Location to store the log file.
      - `dbpath`
        - Directory where you created the database.

   C. Set the size, autoextend, and maxsize values according to your environment. For example, you may want to change the maxsize to a set value rather than UNLIMITED. Consult your Database Administrator for more info.
D. Execute the following SQL script:

**Notes:**
- The DBA or a user with sufficient credentials (CREATE tablespace) must execute the script.
- The script will prompt for the database username and password.

```
# sqlplus
SQL> @jcr_ora_tablespaces.sql
```

Parent topic: Configuring WebSphere Portal to use Oracle RAC
Previous topic: Create users
Next topic: Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Installing Oracle RAC
  - Create users
  - Creating databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:

- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**

- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:

  - `release.DbName` = release
  - `jcr.DbName` = jcrdb
  - `feedback.DbName` = fdbkdb
  - `likeminds.DbName` = lmdb
  - `community.DbName` = commdb
  - `customization.DbName` = custdb

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace `dbdomain` with:
  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If DbUser, DbUrl, and DbPassword are not the same across domains, the value for DataSourceName must differ from the DataSourceName of the other domains. In other words, this value must be unique for the database domain.

If you are using a single-database, single-user, and multi-schema database transfer, the user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.
   A. For dbdomainDbType, type oracle.
   B. For dbdomainDbName, type the name of the WebSphere Portal domain database. Note: This value is also the database element in the dbdomainDbUrl property.
   C. For dbdomainDbSchema, type the schema name of the database domain. Note: Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The dbdomainDbName should be the same value used for the dbdomainDbSchema
      Restriction: The value for dbdomainDbSchema must equal the value for dbdomainDbUser.
   D. For dbdomainDataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   E. For dbdomainDbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. Note: The database element of this value should match the value of DbName.
      - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following: jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.
F. For `dbdomain.DbUser`, type the user ID for the database administrator. **Restriction:** The value for `dbdomain.DbUser` must equal the value for `dbdomain.DbSchema`.

G. For `dbdomain.DbPassword`, type the password for the database administrator.

H. For `dbdomain.DBA.DbUser`, type the database administrator user ID for privileged access operations during creation of the database.

I. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.

J. For `dbdomain.DbHome`, type the root location for the database. **Note:** This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

   A. For `source.domainDbType`, type of the database you are currently configured to use. The value for `source.domainDbType` is Derby by default.

   B. For `source.domainDbName`, type the name of the database domain you are currently using.

   C. For `source.domainSchema`, type current schema identifier for objects within the database for this domain.

   D. For `source.domainDataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

   E. For `source.domainDbUrl`, type the url currently used to access your database.

   F. For `source.domainDbUser`, type the name of the user accessing this database.

   G. For `source.domainDbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

   A. For `oracle.DbDriver`, type the name of the class that SqlProcessor uses to import SQL files.

   B. For `oracle.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For `oracle.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC

**Previous topic:** Creating databases

**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

Before you begin, ensure that the following prerequisites are met:
- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

1. On the database server, make sure that the folder `db_1` contains the subfolders `your_oracle_instance/data` and `your_oracle_instance/index`. If this folder hierarchy does not exist, create it manually before you run the `setup-database` task. The `setup-database` task requires these folders to create database users. If these folders do not exist, the `setup-database` task will fail.

2. Change to the directory `wp_profile_root/ConfigEngine`

3. To create the database users, type the following command:

   ```
   ./ConfigEngine.sh setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:

   `<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql`.

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC  
**Previous topic:** Modifying database properties  
**Next topic:** Creating JCR table spaces

**Related tasks**
Manually creating users and granting privileges for Oracle
Creating JCR table spaces

This topic provides manual instructions for creating JCR table spaces for Oracle.

- **Prerequisites**
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties
  - Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.
2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:
   - &jcrdb is the name of the database you created to store user data.
   - &dbpath is the directory where you created the database; the default path is `/oracle/oradata`.

   Ensure that the `.` is included in the variables when you substitute the values of your environment with these variables.

   **Important:** You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

   ```
   create tablespace ICMLFQ32 datafile 'dbpath./jcrdb/data/jcrdb._ICMLFQ32_01.dbf' size 300M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLNF32 datafile 'dbpath./jcrdb/data/jcrdb._ICMLNF32_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMVFQ04 datafile 'dbpath./jcrdb/data/jcrdb._ICMVFP04_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMSFQ04 datafile 'dbpath./jcrdb/data/jcrdb._ICMSFQ04_01.dbf' size 150M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLSNDX datafile 'dbpath./jcrdb/index/jcrdb._ICMLSNDX_01.dbf' size 10M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
   ```

   A. Set the `size`, `autoextend`, and `maxsize` values according to your environment. For example, you may want to change the `maxsize` to a set value rather than `UNLIMITED`.
   
   B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.
   
   C. Refer to the Oracle command reference for more information about using the `create tablespace` command.

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC  
**Previous topic:** Setting up databases  
**Next topic:** Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:

- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

- Prerequisites
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties
  - Setting up databases
  - Creating JCR table spaces

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.index_name.indexspace
   For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword TABLESPACE and a space. For example: community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE
   Repeat this step for each domain that you are transferring.
4. Save and close dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
- UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC  
**Previous topic:** Creating JCR table spaces  
**Next topic:** Configuring WebSphere Portal to use Oracle RAC
Configuring WebSphere Portal to use Oracle RAC

This section provides information on how to manually transfer data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- Prerequisites
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties
  - Setting up databases
  - Creating JCR table spaces

Tips:
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

- The WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following: jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.
- Manually specify an RAC datasource URL in the WebSphere Application Server console. The URL should look like the following:

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT=1521)) (ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICENAME)).
```

- When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

Steps for transferring data to another supported database
1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.

2. Enter the following commands to validate configuration properties.
   ```
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password
   ```

3. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.

4. Stop both WebSphere Application Server and the WebSphere Portal server:
   ```
   ./stopServer.sh server1 -username admin_userid -password admin_password
   ```

5. Transfer the database:
   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory `wp_profile_root/ConfigEngine`.

   B. Enter the following command:
   ```
   ./ConfigEngine.sh database-transfer -DWasPassword=password
   ```
   **Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:
   ```
   ./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWasPassword=password
   ```

   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

   If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response.
   Example: SQL> execute dbms_stats.gather_schema_stats(ownname=> 'jcr', cascade=> TRUE);

7. Specify the JDBC URL to connect to the cluster:
   A. Login to the WebSphere Application Server Administrator Console

   B. Navigate to Resources > JDBC Providers

   C. If there is a value in the Node field, remove it and click Apply.

   D. For each Oracle JDBC provider, repeat the following steps:

   1. Click the provider name.
   2. Select Data Sources.
   3. Click the name of the data source.
   4. Under the Oracle data source properties, modify the URL attribute according to Oracle RAC configuration:
   ```
   jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT =1521)) (ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521))) (CONNECT_DATA=(SERVICE_NAME= DATABASE_SERVICENAME)))
   ```

   5. Save your changes

8. Change to the directory `wp_profile_root/bin`.

9. Enter the following command to start the WebSphere Portal server:
   ```
   ./startServer.sh WebSphere_Portal
   ```

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC  
**Previous topic:** Assigning custom table spaces
Configuring WebSphere Portal to use SQL Server

View information on installing and setting up SQL Server to work with WebSphere Portal.

1. Installing SQL Server
   View the steps to install SQL Server for use with WebSphere Portal.

2. Modifying database properties
   This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

3. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

4. Optional: Assigning custom filegroups
   The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

5. Configuring WebSphere Portal to use SQL Server 2005
   This section provides information on how to manually transfer data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

**Parent topic:** Configure WebSphere Portal to use a remote database
Installing SQL Server

View the steps to install SQL Server for use with WebSphere Portal.

Before you begin this task, complete the following prerequisites:

- You should have completed reviewing the Planning for SQL Server topic.
- You must install SQL Server separately from WebSphere Portal.
- WebSphere Portal requires either the DataDirect Connect for JDBC 3.7 JDBC driver or the Microsoft SQL Server JDBC Driver.
- You can obtain the DataDirect Driver from DataDirect Technologies. The JDBC driver consists of two parts: the JDBC driver libraries and JTA libraries. The JDBC driver libraries must be put on the WebSphere Portal system, while the JTA libraries are intended for SQL Server.
- You can obtain the Microsoft SQL Server JDBC Driver from Microsoft. See the Microsoft SQL Server product documentation for installation details.

The following section provides instructions for installing SQL Server for use with IBM® WebSphere® Application Server and WebSphere Portal. These steps are the same for both the DataDirect and Microsoft drivers unless noted.

1. Install SQL Server and all required patches.
2. Select the authentication mode for this installation:
   - Windows Authentication Mode
   - Mixed Mode (Windows Authentication and SQL Server Authentication)
     Either authentication mode allows the user who is specified as the database administrator in the wkplc_comp.properties file to log in to the SQL server. Mixed Mode authentication allows either a Windows user or a SQL server user, or both, to log in to the SQL Server.
3. In the SQL Server Setup panel, **Components to Install**, select the following components, which are required services for WebSphere Portal:
   - **SQL Server Database Services**
     - **Integration Services** The option **Integration Services**, creates the directory DTS/Binn, where you will need to copy files when you install Microsoft SQL Server JDBC drivers and enable XA connections.
4. Complete the installation using SQL Server documentation as a guide.
5. Enable TCP/IP connectivity in the **SQL Server Configuration Manager**.
6. Install the JDBC driver using one of these methods:
   - Installing DataDirect Connect for JDBC drivers on UNIX
   - Installing DataDirect Connect for JDBC drivers on Windows
   - Installing Microsoft SQL Server JDBC drivers and enabling XA connections
7. Network the connections for MSDTC in the Windows Component Services as described in Microsoft Help and Support.
8. Start SQL Server.

### Installing DataDirect Connect for JDBC drivers on UNIX

1. Purchase and download DataDirect Connect for JDBC and save file 360connectjdbc.jar in a temporary work directory.
2. To create the required files, run the following command from the directory that contains 360connectjdbc.jar:
   
   ```bash
   jar -xvf 360connectjdbc.jar
   ```

3. Run `./Installer.sh` in the same directory.

4. When installation is complete, change directory to `/installation/lib`, where `installation` is the directory in which you installed the DataDirect Connect for JDBC drivers.

5. Run the following command to change the permissions on the JAR files in the `installation/lib` directory:
   ```bash
   chmod 777 *.jar
   ```

6. Change the ownership and group of these files by running the following commands in the `installation/lib` directory:
   ```bash
   chgrp system_grp *.jar
   chown root *.jar
   ```
   Where `system_grp` is the system group as labeled by your operating system.

### Installing Microsoft SQL Server JDBC drivers and enabling XA connections

1. Download and install the Microsoft SQL Server JDBC driver.

2. Copy file `sqljdbc_xa.dll` from the `xa` subdirectory to the `C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqljdbc_xa.dll` directory of the SQL Server installation.

3. Start the database server.

4. Ensure that the Distributed Transaction Coordinator has been started. The status can be verified in the list of services in the Computer Management console.

5. Start the Microsoft SQL Server Management Studio and connect to the local database engine as the system administrator, `sa`.

6. Select **File > Open > File** and select `xa_install.sql` from the subdirectory of the downloaded and extracted JDBC driver.

7. Execute the script by selecting **Query > Execute**. Note: Any warnings that appear in the messages section of the application window that say that stored procedures cannot be found can be safely ignored.

   
   A. Open the Windows Registry Editor (regedit) and navigate to the element
      
      ```plaintext
      HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSDTC\XADLL
      ```
   
   B. From the menu bar, select **Edit > New > String Value** to create a new parameter named `sqljdbc_xa.dll` in that element.
   
   C. Change the value of the new parameter to the location of the `sqljdbc_xa.dll` file copied in Step 2 above, for example: `C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqljdbc_xa.dll`

---

**Parent topic:** Configuring WebSphere Portal to use SQL Server

**Next topic:** Modifying database properties
Modifying database properties

This section provides information on how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Installing SQL Server

  The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

  **Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
  - WebSphere Portal 6.1.5 wkplc.properties file reference
  - WebSphere Portal 6.1.5 wkplc_comp.properties file reference
  - WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

- **Working with properties files:**
  - The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:
    - `release.DbName=release`
    - `jcr.DbName=jcrdb`
    - `feedback.DbName=fdbkdb`
    - `likeminds.DbName=lmdb`
    - `community.DbName=commdb`
    - `customization.DbName=custdb`  
  - If you are using a remote database, enter the values for the remote server.
  - Use a forward slash (/) instead of a backslash (\).
  - There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
  - The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
  - Depending on which database domain has to be configured, replace `dbdomain` with:
    - `release`
    - `customization`
    - `community`
    - `jcr`
    - `feedback`
    - `likeminds`
  - The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
    - `dbdomain.DbType`
1. Locate the following files and create a backup copy of each before changing any values:
   - `wp_profile_root/ConfigEngine/properties/wkplc.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_comp.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties`

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In `wkplc_comp.properties`, most properties are repeated for each domain.

2. Use a text editor to open the properties file `wkplc_comp.properties` and modify the values to correspond to your environment.
   A. For `dbdomainDbType`, type `sqlserver2005`.
   B. For `dbdomainDbName`, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the `dbdomainDbUrl` property.
   C. For `dbdomainDbSchema`, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   D. For `dbdomainDataSourceName`, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - `releaseDS`
      - `communityDS`
      - `customizationDS`
      - `jcrDS`
      - `lmdbDS`
      - `feedback`
   E. For `dbdomainDbUrl`, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:** The database element of this value should match the value of `DbName`.
   F. For `dbdomainDbUser`, type the user ID for the database administrator.
   G. For `dbdomainDbPassword`, type the password for the database administrator.
   H. For `dbdomainDBA.DbUser`, type the database administrator user ID for privileged access operations during creation of the database.
   I. For `dbdomainDBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.
   J. For `dbdomainDbHome`, type the root location for the database. **Note:** This value is the location to store the database files locally.
   K. For `dbdomainAdminUrl`, type the sqlserver URL without a database attached. This value is used to connect to the server for database administration operations.
   L. For `dbdomainDbHostName`, type the hostname of the database.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are...
set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

A. For source.domainDbType, type of the database you are currently configured to use. The value for source.domainDbType is Derby by default.

B. For source.domainDbName, type the name of the database domain you are currently using.

C. For source.domainDbSchema, type current schema identifier for objects within the database for this domain.

D. For source.domainDataSourceName, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

E. For source.domainDbUrl, type the url currently used to access your database.

F. For source.domainDbUser, type the name of the user accessing this database.

G. For source.domainDbPassword, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file wkplc_dbtype.properties.

A. For sqlserver2005.DbDriver, type the name of the class that SqlProcessor uses to import SQL files.

B. For sqlserver2005.DbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

C. For sqlserver2005.JdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

D. For sqlserver2005.DbConnectionPoolDataSource, type the name of the implementation class of the connection pool data source.

6. Save and close the file.

7. Update the following property in the file wkplc.properties.

A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

Parent topic: Configuring WebSphere Portal to use SQL Server

Previous topic: Installing SQL Server

Next topic: Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

Before you begin, ensure that the following prerequisites are met:
- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

- Prerequisites
  - Installing SQL Server
  - Modifying database properties

1. Change to the directory `wp_profile_root/ConfigEngine`

2. To create the databases, type the following command:
   ```
   ./ConfigEngine.sh create-database -DWasPassword=password
   ```

3. To create the database users, type the following command:
   ```
   ./ConfigEngine.sh setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal: `\<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql`.

**Parent topic:** Configuring WebSphere Portal to use SQL Server

**Previous topic:** Modifying database properties

**Next topic:** Assigning custom filegroups

**Related tasks**

Create users and databases for SQL Server 2005 on AIX and UNIX
Assigning custom filegroups

The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

Before you begin:

- The custom filegroups must exist prior to the execution of the database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties` file.
- For details on creating filegroups refer to the documentation for the database.

**Prerequisites**

- Installing SQL Server
- Modifying database properties
- Setting up databases

If custom filegroups are assigned, each must be assigned explicitly. The default filegroups can be used to contain database objects; however the name of the default file group must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom filegroups:

1. Determine the names of your custom filegroups.
2. Open the mapping file `wp_profile_root/PortalServer/config/tablespaces` that specifies the table space and index space for each property pair for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.indexspace.indexspace`
   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
     - release
     - community
     - customization
     - jcr
     - feedback
     - likeminds
3. Assign a filespace to each entry in the mapping file. The filegroup name must be prepended by the keyword `ON` and a space. For example: `community.COMP_INST.tablespace=ON COMM8KSPACE`
   Repeat this step for each domain that you are transferring.
4. Save and close `dbdomain.space_mapping.properties`.
5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - Windows: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
   - UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
   - i5/OS: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
Configuring WebSphere Portal to use SQL Server 2005

This section provides information on how to manually transfer data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005.

As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

- **Prerequisites**
  - Installing SQL Server
  - Modifying database properties
  - Setting up databases

Tips:
- If you are transferring from Oracle, the open Cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

Steps for transferring data to another supported database

1. Open a command prompt and change to the directory `wp_profile_root/ConfigEngine`.
2. Enter the following commands to validate configuration properties.
   ```
   ./ConfigEngine.sh validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ./ConfigEngine.sh validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ``
3. From the same command prompt as the previous steps, change to the directory `wp_profile_root/bin`.
4. Stop both WebSphere Application Server and the WebSphere Portal server:
   ```
   Option                     Description
   WebSphere Application Server ./stopServer.sh server1 -username admin_userid -password admin_password
   WebSphere Portal           ./stopServer.sh WebSphere_Portal -username admin_userid -password admin_password
   ```
5. Transfer the database:
   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.
   A. Change to the directory `wp_profile_root/ConfigEngine`. 

1888
B. Enter the following commands:

```
./ConfigEngine.sh database-transfer -DWaspPassword=password
```

**Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:

```
./ConfigEngine.sh database-transfer -DTransferDomainList=jcr -DWaspPassword=password
```

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. Change to the directory `wp_profile_root/bin`.

7. Enter the following command to start the WebSphere Portal server:

```
./startServer.sh WebSphere_Portal
```

8. Update the SQL Server 2005 statistics for Portal, and JCR databases by opening SQL Server Management Studio, selecting **New Query**, and running the following query:

```
use db_name
exec sp_updatestats @resample='resample';
```

---

**Parent topic:** Configuring WebSphere Portal to use SQL Server

**Previous topic:** Assigning custom filegroups
Verifying databases

After you configure IBM® WebSphere® Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.

- Prerequisites
- [Technotes for database connectivity issues](#)

You can verify the database connection using IBM WebSphere Application Server or by opening WebSphere Portal in a browser.

- To verify that the WebSphere Portal application server is running by using WebSphere Application Server, complete these steps:
  1. Open the WebSphere Application Server administrative console by entering the following address in a browser:
     
     http://hostname.example.com:10027/ibm/console
     
     where hostname.example.com is the fully qualified host name of the machine where WebSphere Portal is running and 10027 is the default transport port that is created by WebSphere Application Server.
  2. Log into the administrative console.
  3. Depending on your version of WebSphere Application Server, click the appropriate option:
     
     - For WebSphere Application Server Version 6.1: Click Resources & JDBC Providers.
     - For WebSphere Application Server Version 7.0: Click Resources > JDBC > JDBC Providers
  4. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
  5. Select the name of the data source that is defined in wkplc_comp.properties. The default data source is wpdbDS.
  6. Select the name of the JDBC provider that is specified in wkplc_dbtype.properties. The default JDBC provider is wpdbJDBC_dbtype, where dbtype is replaced by the value that matches your environment.
  7. Click Test Connection to verify the database connection. If configuration parameters have been changed, you might need to restart WebSphere Application Server for the test to complete.

- To verify that the WebSphere Portal application server is running by opening WebSphere Portal in a browser, enter the following URL in a supported browser:

  http://hostname.example.com:10040/wps/portal

  where hostname.example.com is the fully qualified host name of the machine where WebSphere Portal is running and 10040 is the default transport port that is created by WebSphere Application Server.

Parent topic: Configure WebSphere Portal to use a remote database
Configuring the primary node to communicate with the deployment manager on Solaris

After installing IBM® WebSphere® Application Server Network Deployment and installing IBM WebSphere Portal on the primary node, you must configure the primary node and the deployment manager to communicate with each other.

- **Prerequisites**
  - Installing WebSphere Portal on Solaris on the primary node
  - Configure WebSphere Portal to use a remote database

Perform the following step to configure WebSphere Portal to communicate with the deployment manager:

Perform the following steps to collect files from the primary node and copy them to the deployment manager:

1. Run the `./ConfigEngine.sh collect-files-for-dmgr -DWasPassword=platformpassword` task, from the `wp_profile_root/ConfigEngine` directory of the primary node, to create an archive or compressed file containing all the files which need to be copied to the Deployment Manager. **Note:** The archive or compressed file will be placed in the `wp_profile_root/filesForDmgr` directory and the file will be called `filesForDmgr.zip`.

2. Stop the deployment manager.

3. Expand the `filesForDmgr.zip` file into the installation root directory of the Deployment Manager; for example in the `/opt/IBM/WebSphere/AppServer` directory. **Note:** If the Deployment Manager profile was not created in the default `AppServer/profiles/Dmgr01` directory, then the `metadata_wkplc.xml` file, located in the `AppServer/profiles/Dmgr01/config/.repository/metadata_wkplc.xml` directory in the zip file, must be placed into the correct Deployment Manager profile directory.

4. If the Deployment Manager profile is running on the same application server where WebSphere Portal is installed, remove the `com.ibm.ws.portletcontainer.deploytask_6.1.5.jar` file from the `AppServer/plugins` directory.

5. Start the deployment manager.

**Parent topic:** Preparing the primary node on Solaris  
**Previous topic:** Configure WebSphere Portal to use a remote database  
**Next topic:** Removing search collections on Solaris

**Related tasks**

- Configuring Portal Search in a cluster on Solaris
Removing search collections on Solaris

If you plan to use search in a cluster, you must configure a remote search server; see Configuring Portal Search in a cluster for information. If you created any search collections, you will need to recreate them on the remote search server. If your search collection has data, export the collection before deleting it and then import the data on the remote server.

- Prerequisites
  - Installing WebSphere Portal on Solaris on the primary node
  - Configure WebSphere Portal to use a remote database
  - Configuring the primary node to communicate with the deployment manager on Solaris

Perform the following step to remove the search collection:

Perform the following steps to delete all existing search collections from the primary node:

1. Log on to WebSphere Portal.
2. Navigate to Administration > Search Administration > Manage Search and then click Search Collections.
3. Click the Delete Collection icon for each search collection and then click OK until they are all deleted.
4. Restart the WebSphere_Portal server and then navigate back to the Search Collections page to verify that all search collections have been deleted.

Parent topic: Preparing the primary node on Solaris
Previous topic: Configuring the primary node to communicate with the deployment manager on Solaris
Choosing the type of cluster to create on Solaris

If you installed a IBM® WebSphere® Application Server Network Deployment, create a static cluster to handle failover requests. If you installed a IBM WebSphere Virtual Enterprise, create a dynamic cluster to balance member workloads.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on Solaris
  - Preparing your Solaris operating system
  - Preparing the primary node on Solaris

Choose one of the following options to create a cluster:

- **Creating a static cluster on Solaris**
  After installing IBM WebSphere Portal on the primary node, configuring a remote database, and preparing the primary node to communicate with the Deployment Manager, you can create your static cluster to handle failover requests.

- **Creating a new dynamic cluster on Solaris using WebSphere Virtual Enterprise**
  After installing and configuring IBM WebSphere Portal on your primary node and all additional nodes, you can create a new dynamic cluster. A dynamic cluster monitors performance and load information and is able to dynamically create and remove cluster members based on the workload.

**Parent topic:** Setting up a cluster on Solaris  
**Previous topic:** Preparing the primary node on Solaris  
**Next topic:** Preparing a remote Web server when portal is installed on Solaris
Creating a static cluster on Solaris

After installing IBM® WebSphere® Portal on the primary node, configuring a remote database, and preparing the primary node to communicate with the Deployment Manager, you can create your static cluster to handle failover requests.

Perform the following steps to create the cluster:

1. If the Deployment Manager is configured to use a stand-alone LDAP user registry, update the wkplc.properties file, located in the \wp_profile_root\ConfigEngine\properties directory, on the primary node with the stand-alone LDAP user registry property values from the Deployment Manager. You can find these settings under the VMM Stand-alone LDAP configuration heading.

   **Important:** Ensure that you set **WasUserid** and **WasPassword** to the Deployment Manager user ID and password.

2. Stop the server1 and WebSphere_Portal servers on the primary node and ensure that the following parameters are set correctly in the wkplc.properties file, located in the \wp_profile_root\ConfigEngine\properties directory:

   **Note:** Although you can add these parameters directly to any task that you run while creating your cluster, you may want to add them to the properties file while creating your cluster and then remove them when you are finished to keep your environment secure.

   A. Set **WasSoapPort** to the port used to connect remotely to the deployment manager.
   B. Set **WasRemoteHostName** to the full host name of the server used to remotely connect to the deployment manager.
   C. Verify that **WasPassword** is set to your deployment manager password.
   D. Verify that **PortalAdminPwd** is set to your WebSphere Portal password.
   E. Verify that **ClusterName** is set.
   F. Verify that **PrimaryNode** is set to **true**.

3. Make a backup copy of the \wp_profile_root\config\cells\cell_name\wim\config\wimconfig.xml and \wp_profile_root\config\cells\cell_name\wim\model\wimxmlextension.xml, if available, files.

4. Run the ./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=dmgr_password task, from the \wp_profile_root\ConfigEngine directory of the primary node.

   **Note:** If you want to specify custom ports for the nodeagent, add the -DPortPropsFile=full to portsfile parameter to the cluster-node-config-pre-federation task. You can use the ports files that are found on the Setup CD for WebSphere_Portal and server1 as a guide.

   **Note:** You may receive a message about accepting an SSL signer certificate. Failure to accept the SSL signer certificate will cause the script to fail. Alternatively, the com.ibm.ssl.enableSignerExchangePrompt flag can be enabled in the ssl.client.props file for "DefaultSSLSessions" in order to allow acceptance of the signer during the connection attempt.

   **Warning:** If the cluster-node-config-pre-federation fails for any reason, you must perform the following steps before rerunning the task:

   A. Remove the node if the AddNode task succeeded.
   B. Log on to the deployment manager and perform the following steps if the items exist:
      1. Remove all enterprise applications.
      2. Remove the WebSphere_Portal server definition.
      3. Remove the WebSphere Portal JDBC Provider.

   **Note:** If you are migrating from a version prior to Version 6.1.0 that contains Lookaside data, read the following technote: Migrating with Lookaside Data.
Optional: If you configured a database user registry or a property extension (lookaside) database, run the following task to set up access to the database drivers:

**Note:** You will also need to run this task if you migrated the primary node from a release prior to Version 6.1.0, even if you did not configure a database user registry or a property extension (lookaside) database.

A. Set the property value for `federated.dbDbType` if using a database user registry or set the property value for `laDbType` if using a property extension database in the `wkplc.properties` file.

**Note:** If you migrated the primary node from a version prior to Version 6.1.0 and you are not using a database user registry or a property extension database, set the property value for `federated.dbDbType` to the same value that is in the `wkplc.properties` file on the primary node.

B. Run the `./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWassPassword=dmgr_password -DDBDomain=la|federated.db -DVmmNodeName=node_name -Db_type.NodeDbLibrary=local full path of the database jars` task from the `wp_profile_root`/ConfigEngine directory to create the variable used to access the VMM database jars.

**Note:** The `VmmNodeName` is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The `db_type` in `db_type.NodeDbLibrary` should be set to the type of database you are using, for example `db2`.

C. Run the `./ConfigEngine.sh cluster-node-config-post-federation -DWassPassword=dmgr_password` task.

Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell's user registry. Run the `./ConfigEngine.sh wp-change-portal-admin-user -DWassPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task, from the `wp_profile_root`/ConfigEngine directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

**Attention:** The password value for `-DWassPassword` is the Deployment Manager administrative password.

**Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

**Fast path:** If the value for `newAdminGroupId` contains a space; for example `Software Group`, open the `wkplc.properties` file and add the values for `newAdminId`, `newAdminPw`, and `newAdminGroupId`. Save your changes and run the `ConfigEngine.bat wp-change-portal-admin-user -DWassPassword=dmgr_password` task.

**Important:** If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user` task until after the `cluster-node-config-cluster-setup` (static cluster) or `cluster-node-config-dynamic-cluster-setup` (dynamic cluster) task completes. After running the `wp-change-portal-admin-user` task, start or restart the WebSphere Portal server to use the updated administrator user ID.

**Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the `wp-change-portal-admin-user` task. `-DnewAdminPw` is an optional parameter to update the Administrative password in the `wkplc.properties` file if required.

D. Run the `./ConfigEngine.sh cluster-node-config-cluster-setup -DWassPassword=dmgr_password` task.

Configure the cluster to use an external Web server to take advantage of features such as workload management. Choose one of the following options:

- Configuring a Web server and an application server on separate machines (remote)
- Configuring a Web server and an application server profile on the same machine
- Configuring a Web server and a deployment manager profile on the same machine

**Note:** Start with the step about launching the Plug-ins installation wizard.

E. Perform the following steps to access the Web Content Management content through an external Web server:

A. Log on to the deployment manager administrative console.
B. Select Environment > WebSphere Variables.
C. From the Scope drop-down menu, select the Node=nodename, Server=servername option to narrow the scope of the listed variables, where Node=nodename is the node that contains the application server.
D. Update the WCM_HOST variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.
E. Update the WCM_PORT variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.
F. Perform the following steps to synchronize the node with the deployment manager:
   1. Select System Administration > Nodes.
   2. Select the node that you want to synchronize from the list.
   3. Click Full Resynchronize.
G. Log off of the deployment manager administrative console.

11. Run the following tasks to propagate your changes: Note: WebSphere_Portal_nodename is the name of the node’s WebSphere Portal server; but if you customized the server name, the default name changes. If you are uncertain of the server name, run the serverstatus -all task to get a list of the server names and their status.
A. ./stopManager.sh -username admin_userid -password admin_password, from the dmgr_profile_root\bin directory
B. ./stopNode.sh -username admin_userid -password admin_password, from the wp_profile_root/bin directory
C. ./stopServer.sh WebSphere_Portal_nodename -username admin_userid -password admin_password
D. ./startManager.sh, from the dmgr_profile_root\bin directory
E. ./startNode.sh, from the wp_profile_root/bin directory
F. ./startServer.sh WebSphere_Portal_nodename

12. Change the task list settings to point to the cluster instead of a single server if using Portal business process integration. See one of the following sections in the Information Center for information:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross cell</td>
<td>Adding a BPI-enabled portal server to a managed cell in a cross-cell setup</td>
</tr>
<tr>
<td>Single cell</td>
<td>Adding a BPI-enabled portal server to a managed cell in a single-cell setup</td>
</tr>
</tbody>
</table>

Parent topic: Choosing the type of cluster to create on Solaris

Related tasks
- Adding a BPI-enabled portal server to a managed cell in a cross-cell setup
- Adding a BPI-enabled portal server to a managed cell in a single-cell setup
Creating a new dynamic cluster on Solaris using WebSphere Virtual Enterprise

After installing and configuring IBM® WebSphere® Portal on your primary node and all additional nodes, you can create a new dynamic cluster. A dynamic cluster monitors performance and load information and is able to dynamically create and remove cluster members based on the workload.

Before creating your dynamic cluster, install and configure the Deployment Manager and perform all the tasks under “Preparing the primary node.” On the Deployment Manager system, install WebSphere Virtual Enterprise and augment the deployment manager profile. On all WebSphere Portal nodes, install WebSphere Virtual Enterprise and augment the wp_profile profile. See Installing and configuring the product for information about installing WebSphere Virtual Enterprise. See Augmenting profiles for information about augmenting the Deployment Manager and wp_profile profiles.

- Prerequisites
  - PK97393 for all WebSphere Application Server Versions prior to 6.1.0.5
  - Installing and configuring the product:
    - Augmenting profiles

Perform the following steps to create a dynamic cluster:

1. Perform the following steps for on the primary node of the dynamic cluster:
   A. Prepare the node for the dynamic cluster; perform all tasks under Preparing the primary node.
   B. If the Deployment Manager is configured to use a stand-alone LDAP user registry, update the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory, on the primary node with the stand-alone LDAP user registry property values from the Deployment Manager. You can find these settings under the VMM Stand-alone LDAP configuration heading.
   C. Stop the server1 and WebSphere_Portal servers on the primary node and ensure that the following parameters are set correctly in the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory:
      1. Set WasSoapPort to the port used to connect remotely to the deployment manager.
      2. Set WasRemoteHostName to the full host name of the server used to remotely connect to the deployment manager.
      3. Verify that WasPassword is set to your deployment manager password.
      4. Verify that PortalAdminPwd is set to your WebSphere Portal password.
      5. Verify that ClusterName is set.
      6. Verify that PrimaryNode is set to true.
   D. Run the ./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=dmgr_password task, from the wp_profile_root/ConfigEngine directory of the primary node.

   Warning: If the cluster-node-config-pre-federation fails for any reason, you must perform the following steps before rerunning the task:
      1. Remove the node if the AddNode task succeeded.
      2. Log on to the deployment manager and perform the following steps if the items exist:
         A. Remove all enterprise applications.
         B. Remove the WebSphere_Portal server definition.
         C. Remove the WebSphere Portal JDBC Provider.

Note: If you are migrating from a version prior to Version 6.1.0 that contains Lookaside data, read the following
E. Run the following task. Include each node name as a comma separated list in the command:

1. Set the property value for `federated.db.DbType` if using a database user registry or set the property value for `la.DbType` if using a property extension database in the `wkplc.properties` file. **Note:** If you migrated the primary node from a version prior to Version 6.1.0 and you are not using a database user registry or a property extension database, set the property value for `federated.db.DbType` to the same value that is in the `wkplc.properties` file on the primary node.

2. Run the `./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=dmgr_password -DDomainName=la|federated.db -DVmmNodeName=node_name -Db_type.NodeDbLibrary=local full path of the database jars` task from the `wp_profile_root/ConfigEngine` directory to create the variable used to access the VMM database jars. **Note:** `VmmNodeName` is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The `db_type` in `db_type.NodeDbLibrary` should be set to the type of database you are using, for example `db2`.

F. Run the `./ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=dmgr_password task`.

G. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell's user registry. Run the `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task, from the `wp_profile_root/ConfigEngine` directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

**Important:** If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user` task until after the `cluster-node-config-dynamic-cluster-setup` task completes.

After running the `wp-change-portal-admin-user` task, start or restart the WebSphere_Portal server to use the updated administrator user ID.

**Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the `wp-change-portal-admin-user` task. `-DnewAdminPw` is an optional parameter to update the Administrative password in the `wkplc.properties` file if required.

2. Log on to the deployment manager administrative console.

3. Perform the following steps to create a node group:
   A. Click **System administration** > **Node groups**.
   B. Click **New**.
   C. Type the node group **Name**.
   D. Optional: Type any information about the node group in the **Description** text box.
   E. Click **OK**.
   F. Click the **Save** link to save your changes to the master configuration.

4. Perform the following steps to add members to the node group:
   A. Click **System administration** > **Node groups**.
   B. Click on the name of the node group that you want to add members to.
   C. Click **Node group members** under Additional Properties.
   D. Click **Add**.
   E. Select the primary node and then click **Add**.
   F. Click the **Save** link to save your changes to the master configuration.

5. Perform the following steps to create a dynamic cluster in the node group:
   A. Click **Servers** > **Dynamic clusters**.
   B. Click **New**.
C. Select WebSphere Application Server from the **Server Type** pull-down menu and then click **Next**.

D. Type the cluster name in the **Dynamic cluster name** text box and then click **Next**. Type the same value that you provided for the **ClusterName** parameter in the wkplc.properties file of your primary node.

E. Remove all default membership policies and then click **Subexpression builder**.

F. Enter the following information in the Subexpression builder window:

1. Select **AND** from the **Logical operator** pull-down menu.
2. Select **Nodegroup** from the **Select operand** pull-down menu.
3. Select **Equals (=)** from the **Operator** pull-down menu.
4. Type the nodegroup name you created in the previous step in the **Value** text box.
5. Click **Generate subexpression**.
6. Click **Append**.

G. Click **Preview membership** to verify that all nodes included in the nodegroup display and then click **Next**.

H. Click the **Create the cluster member using an existing server as a template** radio button and then select the WebSphere_Portal server for the primary node from the pull-down menu.

I. Click **Next**.

J. Specify the dynamic cluster properties and then click **Next**.

K. Review the summary page to verify your actions and then click **Finish**.

L. Click the **Save** link to save your changes to the master configuration.

6. Define or verify the following parameters in the wkplc.properties file, located in the `wp_profile_root`/ConfigEngine/properties directory:

   A. Set **ClusterName** to the name of the new dynamic cluster.
   
   B. Verify that **CellName** is set to the deployment manager cell.
   
   C. Verify that **NodeName** is set to the local WebSphere Portal node.
   
   D. Set **ServerName** to the server that will be used for the dynamic cluster member on this node.

   **Note:** Log on to the deployment manager administrative console and click Dynamic Clusters > PortalCluster > Dynamic cluster members to find the name of the server used for the dynamic cluster.

E. Verify that **PrimaryNode** is set to true.

7. Run the `./ConfigEngine.sh cluster-node-config-dynamic-cluster-setup -DWasPassword=dmgr_password` task from the `wp_profile_root`/ConfigEngine directory to create the dynamic cluster.

   **Note:** This task may display a warning message since the cluster already exists but it will proceed to create the new dynamic cluster; therefore, the warning can be ignored.

8. Perform the following steps to access the Web Content Management content through an external Web server:

   A. Log on to the deployment manager administrative console.
   
   B. Select Environment > WebSphere Variables.
   
   C. From the *Scope* drop-down menu, select the Node-nodename, Server-servername option to narrow the scope of the listed variables, where Node-nodename is the node that contains the application server.
   
   D. Update the **WCM_HOST** variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.
   
   E. Update the **WCM_PORT** variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.
   
   F. Perform the following steps to synchronize the node with the deployment manager:

   1. Select System Administration > Nodes.
   
   2. Select the node that you want to synchronize from the list.
   
   3. Click Full Resynchronize.
   
   G. Log off of the deployment manager administrative console.

9. Run the following tasks to propagate your changes: **Note:** WebSphere_Portal_nodename is the name of the node's WebSphere Portal server; but if you customized the server name, the default name changes. If you are uncertain of the
server name, run the serverstatus -all task to get a list of the server names and their status.

A. `./stopManager.sh -username admin_userid -password admin_password`, from the dmgr_profile_root/bin directory
B. `./stopNode.sh -username admin_userid -password admin_password`, from the wp_profile_root/bin directory
C. `./stopServer.sh WebSphere_Portal_nodename -username admin_userid -password admin_password`
D. `./startManager.sh`, from the dmgr_profile_root/bin directory
E. `./startNode.sh`, from the wp_profile_root/bin directory
F. `./startServer.sh WebSphere_Portal_nodename`

Parent topic: Choosing the type of cluster to create on Solaris
Preparing a remote Web server when portal is installed on Solaris

Install and configure the Web server plug-in, provided by IBM® WebSphere® Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

- **Prerequisites**
  - Preparing the WebSphere Application Server Deployment Manager on Solaris
  - Preparing prerequisite and corequisite software on Solaris
  - Preparing your Solaris operating system
  - Preparing the primary node on Solaris

Perform the following steps to install and configure your Web server:

1. Install and configure the Web server; refer to the Web server documentation for information.
2. If using IBM Lotus® Domino®, edit the `NOTES.INI` file on the Web server. Set the `HTTPEnableConnectorHeaders` parameter to 1.
3. If using IBM HTTP Server or Apache Server, edit the `httpd.conf` file on the Web server. Set the `AllowEncodedSlashes` directive to `On`; the directive should be added at the root level as a global directive.
4. Stop the Web server.
5. Install and configure the Web server plug-in on the system where the Web server is located using the Plug-ins installation wizard provided with WebSphere Application Server. Refer to Selecting a Web server topology diagram and roadmap for information. **Note:** The WebSphere Application Server Web server instructions are for the default profile. If not using the default profile, refer to Configuring a Web server for a non-default profile.

   If using WebDAV: After successfully installing the Web server plug-in, locate and open your `plugin-cfg.xml` file and set `AcceptAllContent` to `true`.

   **Important:** Depending on how you use the Web server, you may need to adjust the `ServerIOTimeout` value, which defines how long the plug-in should wait for a response from the application. The recommended minimum value is 60 but you may need to adjust this value higher if you are retrieving data from a database. To update this value, locate and open your `plugin-cfg.xml` file and set `ServerIOTimeout` to a value that is appropriate for your business needs. For additional information, see Common questions about the Web server plug-in.
6. If you are using Sun Web Server Version 7 update 8, read Technote 1448262 and perform the steps to resolve the HTTP 408/409 error.
7. Start the Web server.

**Parent topic:** Preparing prerequisite and corequisite software on Solaris

**Previous topic:** Preparing the WebSphere Application Server Deployment Manager on Solaris

**Parent topic:** Setting up a cluster on Solaris

**Previous topic:** Choosing the type of cluster to create on Solaris

**Next topic:** Configuring WebSphere Portal to use a user registry on Solaris in a clustered environment
Configuring WebSphere Portal to use a user registry on Solaris in a clustered environment

Configure user registry security on IBM® WebSphere® Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

Prior to configuring security, you should use the IBM WebSphere Application Server backupConfig task to create and store a backup of the IBM WebSphere Portal configuration; see backupConfig command for information.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on Solaris
  - Preparing your Solaris operating system
  - Preparing the primary node on Solaris

Perform the following tasks to configure WebSphere Portal to use a user registry:

1. **Preparing user registries on Solaris**
   Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

2. **Choosing your user registry model on Solaris in a clustered environment**
   Choose between securing IBM WebSphere Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.
Preparing user registries on Solaris

Install and setup an LDAP server as a user registry to store user information and authenticate users in your high availability production environment.

Choose the appropriate LDAP server to install and setup:

- **Preparing a Domino Directory server**
  If you plan to use a Domino Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a SecureWay Security Server**
  If you plan to use a SecureWay Security Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Novell eDirectory**
  If you plan to use a Novell eDirectory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Sun Java System Directory Server**
  If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

- **Preparing a Tivoli Directory Server**
  If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM WebSphere Portal.

**Parent topic:** Configuring WebSphere Portal to use a user registry on Solaris in a clustered environment  
**Next topic:** Choosing your user registry model on Solaris in a clustered environment
Preparing a Domino Directory server

If you plan to use a Domino Directory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Domino Directory:

1. Perform the following steps to install the Domino Directory:
   A. Go to Lotus Domino documentation.
   B. Select the appropriate version tab for your product.
   C. Click the Download/View online link for the Lotus Domino for multiple platforms Information Center.
   D. Click Domino Administrator Help > Installation > Installing and setting up Domino servers > Server installation > Installing Domino on Unix and perform this task.
   E. Click Domino Administrator Help > Installation > Installing and setting up Domino servers > The Domino server setup program and perform this task.

2. Perform the following steps as a guide to create the WebSphere Portal administrative user:
   A. Navigate to the People view of the Domino Directory and then click Add Person.
   B. Enter the following values in the New Person form to create the wpsbind user:
      - Last Name
        - wpsbind
      - User Name
        - wpsbind/DominoDomain, where DominoDomain is your Lotus Domino Internet domain
        - wpsbind Note: Make sure you enter two values in the User Name field, where the first value includes the Lotus Domino domain.
      - Short name/UserID
        - wpsbind
      - Internet password
        - wpsbind
   C. Click Save and Close to save the new person record for wpsbind and return to the People view.
   D. Click Add Person and enter the following values in the New Person form to create the wpsadmin user:
      - Last Name
        - wpsadmin, where wpsadmin in the user ID for the WebSphere Portal Administrator
      - User Name
        - wpsadmin/DominoDomain, where DominoDomain is your Lotus Domino Internet domain
        - wpsadmin Note: Make sure you enter two values in the User Name field, where the first value includes the Lotus Domino domain.
      - Short name/UserID
        - wpsadmin
      - Internet password
        - wpsadmin
   E. Click Save and Close to save the new person record for wpsadmin and return to the People view.
   F. Navigate to the Groups view and click Add Group.
   G. Enter the following values in the New Group form on the Basic tab:
      - Group name
- wpsadmins

**Note:** If you plan to configure WebSphere Portal for multiple user registries and your Lotus Domino LDAP will share a realm with another user registry, you must use the hierarchical naming convention for the group names, for example: wpsadmins/DominoDomain, to avoid unexpected results during WebSphere Portal runtime.

- **Group type**
  - Multi-purpose

- **Members**
  - wpsbind/DominoDomain
  - wpsadmin/DominoDomain

**Note:** You can add additional administrator users if required.

H. **Click Save and Close** to save the wpsadmins group with the wpsbind and wpsadmin users as members.

3. Perform the following steps to update the access control list for the Domino Directory:
   
   A. Open the names.nsf file in the Lotus Domino Administrator or Lotus Notes client.
   
   B. **Click File > Application > Access Control** from the main menu to open the access control list for the file.
   
   C. In the **Access Control List > Basics** panel, ensure that the wpsadmins group has either Author or Editor access.
   
   D. Add the following Role Types to the wpsadmins group:
      
      - GroupCreator
      - GroupModifier
      - UserCreator
      - UserModifier
   
   E. **Click OK.**

**Parent topic:** Preparing user registries on Solaris
Preparing a SecureWay Security Server

If you plan to use a SecureWay Security Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare SecureWay Security Server:

1. Install SecureWay Security Server; refer to the IBM SecureWay™ Security Server for z/OS® and OS/390® for information.

2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.

   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.

   C. Replace every dc=yourco,dc=com with your suffix.

   D. Replace any prefixes and suffixes that are unique to your LDAP server.

   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.

   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.

   G. Save your changes.

   H. Follow the instructions provided with your directory server to import the LDIF file.

   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on Solaris
Preparing a Novell eDirectory

If you plan to use a Novell eDirectory as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Novell eDirectory:

1. Install Novell eDirectory; refer to eDirectory for information.

2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every dc=yourco,dc=com with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.
   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on Solaris
Preparing a Sun Java System Directory Server

If you plan to use a Sun Java System Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Sun Java System Directory Server:

1. Install Sun Java System Directory Server; refer to Oracle software for information.

2. Perform the following steps to create the WebSphere Portal administrative user:
   A. Optional: Perform the following steps to create a new directory suffix:
      1. Click the Server Administration folder, located in the left-hand navigation of the directory server console.
      2. Click the Manage Server Properties folder under the Server Administration folder and then select Suffixes on the main page.
      3. Type the Base DN name for the suffix; for example: dc=yourcompany,dc=com.
      4. Click Add.
      5. Click OK to save your changes.
      6. Stop and restart the LDAP server.
   B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
      - Use the PortalUsers.ldif file as a working example and adapted appropriately to work with your LDAP server.
      - Use the ContentUsers.ldif file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   C. Replace every dc=yourco,dc=com with your suffix.
   D. Replace any prefixes and suffixes that are unique to your LDAP server.
   E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the objectclasses to accessGroup. If using Tivoli Access Manager Version 6, set the objectclasses to groupOfNames.
   G. Save your changes.
   H. Follow the instructions provided with your directory server to import the LDIF file.
   I. Stop and restart the LDAP server.

Parent topic: Preparing user registries on Solaris
Preparing a Tivoli Directory Server

If you plan to use a Tivoli Directory Server as an LDAP user registry, you must install and set up the server so that it will communicate with IBM® WebSphere® Portal.

Perform the following steps to prepare Tivoli Directory Server:

   - **Note:** Due to a restriction in Tivoli Directory Server, users or groups must not contain a Turkish uppercase dotted I or lowercase dotted i in the DN as this will prevent correct retrieval of that user or group.

2. Perform the following steps to create the WebSphere Portal administrative user:
   - A. Optional: Perform the following steps to create a new directory suffix:
     1. Click the **Server Administration** folder, located in the left-hand navigation of the directory server console.
     2. Click the **Manage Server Properties** folder under the Server Administration folder and then select **Suffixes** on the main page.
     3. Type the **Base DN** name for the suffix; for example: `dc=yourcompany,dc=com`.
     4. Click **Add**.
     5. Click **OK** to save your changes.
     6. Stop and restart the LDAP server.
   - B. Open the appropriate LDIF file, located in the root directory of the CD setup, with a text editor:
     - Use the `PortalUsers.ldif` file as a working example and adapted appropriately to work with your LDAP server.
     - Use the `ContentUsers.ldif` file for the IBM DB2® Content Manager group and user IDs if you configured DB2 Content Manager.
   - C. Replace every `dc=yourco,dc=com` with your suffix.
   - D. Replace any prefixes and suffixes that are unique to your LDAP server.
   - E. You can specify user names other than wpsadmin and wpsbind. For security reasons, specify nontrivial passwords for these administrator accounts.
   - F. Optional: If using IBM Tivoli® Access Manager Version 5.1, set the **objectclasses** to **accessGroup**. If using Tivoli Access Manager Version 6, set the **objectclasses** to **groupOfNames**.
   - G. Save your changes.
   - H. Follow the instructions provided with your directory server to import the LDIF file.
   - I. Stop and restart the LDAP server.

---

**Parent topic:** Preparing user registries on Solaris
Choosing your user registry model on Solaris in a clustered environment

Choose between securing IBM® WebSphere® Portal with a standalone LDAP user registry or by adding LDAP user registries and/or database user registries to the default federated repository.

Choose one of the following user registry models:

- **Configuring a stand-alone LDAP user registry on Solaris in a clustered environment**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Configuring the default federated repository on Solaris in a clustered environment**
  Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

- **Adapting the attribute configuration**
  After installing IBM WebSphere Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

**Parent topic:** Configuring WebSphere Portal to use a user registry on Solaris in a clustered environment

**Previous topic:** Preparing user registries on Solaris
Configuring a stand-alone LDAP user registry on Solaris in a clustered environment

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

Choose one of the following options to configure your LDAP server:

- **Configuring a stand-alone LDAP user registry on Solaris in a clustered environment**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry to store all user account information for authorization.

- **Configuring a stand-alone LDAP user registry over SSL on Solaris in a clustered environment**
  Configure IBM WebSphere Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

**Parent topic:** Choosing your user registry model on Solaris in a clustered environment
Configuring a stand-alone LDAP user registry on Solaris in a clustered environment

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry to store all user account information for authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps. If you need to rerun the wp-modify-ldap-security task to change the LDAP repositories or because the task failed, you must choose a new name for the realm using the standalone.ldap.realm parameter or you can set ignoreDuplicateIDs=true in the wkplc.properties file, before rerunning the task.

Perform the following steps to configure a standalone LDAP user registry:

Note: Use the wp_security_xxx.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_security_xxx.properties helper file.

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

2. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM Standalone LDAP configuration heading:

   - standalone.ldap.id
   - standalone.ldap.host
   - standalone.ldap.port
   - standalone.ldap.bindDN
   - standalone.ldap.bindPassword
   - standalone.ldap.ldapServerType
   - standalone.ldap.userIdMap
   - standalone.ldap.groupIdMap
   - standalone.ldap.groupMemberIdMap
   - standalone.ldap.userFilter
   - standalone.ldap.groupFilter
   - standalone.ldap.serverId
   - standalone.ldap.serverPassword
   - standalone.ldap.realm
   - standalone.ldap.primaryAdminId
   - standalone.ldap.primaryAdminPassword
   - standalone.ldap.primaryPortalAdminId
   - standalone.ldap.primaryPortalAdminPassword
   - standalone.ldap.primaryPortalAdminGroup
   - standalone.ldap.baseDN
3. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.idap.et.group.objectClasses`
   - `standalone.idap.et.group.objectClassesForCreate`
   - `standalone.idap.et.group.searchBases`
   - `standalone.idap.et.personaccount.objectClasses`
   - `standalone.idap.et.personaccount.objectClassesForCreate`
   - `standalone.idap.et.personaccount.searchBases`

4. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attributes heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.idap.gm.groupMemberName`
   - `standalone.idap.gm.objectClass`
   - `standalone.idap.gm.scope`
   - `standalone.idap.gm.dummyMember`

5. Required: Enter a value for the following required relative distinguished name (RDN®) parameters in the `wkplc.properties` file under the Default parent, RDN attribute heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.idap.personAccountParent`
   - `standalone.idap.groupParent`
   - `standalone.idap.personAccountRdnProperties`
   - `standalone.idap.groupRdnProperties`

6. Save your changes to the `wkplc.properties` file.

7. Optional: Enter the following Web content authors parameters in the `wkplc_comp.properties` file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. **Note:** See the `wkplc_comp.properties` file for specific information about the required parameters and for advanced parameters.
   - `WcmContentAuthorsGroupId`
   - `WcmContentAuthorsGroupCN`

8. Run the `./ConfigEngine.sh validate-standalone-ldap -DWasPassword=password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, **WasPassword** is the value entered during installation and not a value found in your LDAP user registry. **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press y then Enter.

9. Run the `./ConfigEngine.sh wp-modify-ldap-security -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to set the stand-alone LDAP user registry.

10. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

11. Run the `./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

12. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the
LDAP directory.

**Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

A. Edit the `wp_profile_root` file:

```
/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties
```

B. Add the following lines to the file:

```
uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN

cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
```

Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `./ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured: *Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management*

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

13. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring a stand-alone LDAP user registry on Solaris in a clustered environment

**Related tasks**

- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation

---

1915
Configuring a stand-alone LDAP user registry over SSL on Solaris in a clustered environment

Configure IBM® WebSphere® Portal to use a standalone LDAP user registry over SSL to store all user account information for secure authorization.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to configure a standalone LDAP user registry over SSL:

Note: Use the wp_security_XXX.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_security_XXX.properties helper file.

1. To specify the LDAP server's SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:

   - **Server trust store**
     A. Add the certificate to the trust store:
        1. Log in to the WebSphere Application Server Administrative Console.
        2. Navigate to Security > SSL certificate and key management > SSL configurations.
        3. Click the appropriate SSL configuration from the list. For example,
           - Stand-alone environments: NodeDefaultSSLSettings
           - Clustered environments: CellDefaultSSLSettings

           **Clustered environments:** Ensure the setting for SSL configuration for outbound connection matches your SSL settings.

        4. Click Key stores and certificates.
        5. Click the appropriate trust store from the list; for example, NodeDefaultTrustStore.
        6. Click Signer certificates, click Add, and then enter the following information:
           - Type the Alias the key store uses for the signer certificate.
           - Type the File name where the signer certificate is located.
        7. Click OK and then click Save to save the changes to the master configuration.

   B. Retrieve the certificate from the port:
      1. Log in to the WebSphere Application Server Administrative Console.
      2. Navigate to Security > SSL certificate and key management > SSL configurations.
      3. Click the appropriate SSL configuration from the list. For example,
         - Stand-alone environments: NodeDefaultSSLSettings

1916
- Clustered environments: **CellDefaultSSLSettings**

Clustered environments: Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.

4. Click **Key stores and certificates**.

5. Click the appropriate trust store from the list; for example, **NodeDefaultTrustStore**.

6. Click **Signer certificates**, click **Retrieve from port**, and then enter the following information:
   - Type the **Host** name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the **Port** used when attempting to retrieve the signer certificate.
   - Type the **Alias** the key store uses for the signer certificate.

7. Click **Retrieve signer information** to retrieve the certificate from the port.

8. Click **OK** and then click **Save** to save the changes to the master configuration.

- **Client trust store**

  **Note**: This task might report an error, but it does successfully update the trust store. You can ignore the error message.

  A. See Secure installation for client signer retrieval.

  B. Run the **retrieveSigners** task from the **wp_profile_root/bin** directory; see **retrieveSigners command** for information. In a deployed environment, you will need to run the retrieveSigners task, for any federated node, against the Deployment Manager. **Note**: This task might report an error, but it does successfully update the trust store. You can ignore the error message.

  C. Update the trust store properties file:

    1. Use a text editor to open the **ssl.client.props** file, located in the **wp_profile_root/properties** directory.
    2. Change the **com.ibm.ssl.trustStore** parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter
       
       ```
       com.ibm.ssl.trustStore=${CONFIG_ROOT}\cells\wpsbvt\nodes\wpsbvt\trust.p12
       ```
       
       to use the default trust store.
    3. Save your changes.

  2. Use a text editor to open the **wkplc.properties** file, located in the **wp_profile_root/ConfigEngine/properties** directory.

  3. Required: Enter a value for the following required parameters in the **wkplc.properties** file under the VMM Standalone LDAP configuration heading. **Note**: See the properties file for specific information about the required parameters and for advanced parameters.

    - **standalone.idap.ld**
    - **standalone.idap.host**
    - **standalone.idap.port**
    - **standalone.idap.blIndDN**
    - **standalone.idap.bindPassword**
    - **standalone.idap.idapServerType**
    - **standalone.idap.userIdMap**
    - **standalone.idap.groupIdMap**
    - **standalone.idap.groupMemberIdMap**
    - **standalone.idap.userFilter**
    - **standalone.idap.groupFilter**
    - **standalone.idap.serverId**
    - **standalone.idap.serverIdPassword**
    - **standalone.idap.realm**
    - **standalone.idap.primaryAdminId**
    - **standalone.idap.primaryAdminPassword**
    - **standalone.idap.primaryPortalAdminId**
4. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.ldap.et.group.objectClasses`
   - `standalone.ldap.et.group.objectClassesForCreate`
   - `standalone.ldap.et.group.searchBases`
   - `standalone.ldap.et.personaccount.objectClasses`
   - `standalone.ldap.et.personaccount.objectClassesForCreate`
   - `standalone.ldap.et.personaccount.searchBases`

5. Required: Enter a value for the following required group member parameters in the `wkplc.properties` file under the Group member attributes heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.ldap.gm.groupMemberName`
   - `standalone.ldap.gm.objectClass`
   - `standalone.ldap.gm.scope`
   - `standalone.ldap.gm.dummyMember`

6. Required: Enter a value for the following required relative distinguished name (RDN®) parameters in the `wkplc.properties` file under the Default parent, RDN attribute heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `standalone.ldap.personAccountParent`
   - `standalone.ldap.groupParent`
   - `standalone.ldap.personAccountRdnProperties`
   - `standalone.ldap.groupRdnProperties`

7. Enter a value for the following parameters to enable Secure Socket Layers (SSL): **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   Required parameters:
   - `standalone.ldap.sslEnabled`
   - `standalone.ldap.sslConfiguration`

   Optional parameters:
   - `standalone.ldap.certificateMapMode`
   - `standalone.ldap.certificateFilter`

8. Save your changes to the `wkplc.properties` file.

9. Run the `./ConfigEngine.sh validate-standalone-ldap -DWasPassword=password` task to validate your LDAP server settings. **Attention:** If you have not deleted the default file repository, `WasPassword` is the value entered during installation and not a value found in your LDAP user registry. **Note:** During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press `y` then `Enter`.

10. Run the `./ConfigEngine.sh wp-modify-ldap-security -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to set the stand-alone LDAP user registry.

11. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

12. Run the `./ConfigEngine.sh wp-validate-standalone-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured
LDAP user registry.**Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring a stand-alone LDAP user registry on Solaris in a clustered environment

**Related tasks**
- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation
Configuring the default federated repository on Solaris in a clustered environment

Add an LDAP user registry and/or a database user registry to the default federated repository to create seamless authentication within the repository.

Perform the following tasks to add user registries to the default federated repository:

- **Configuring a federated LDAP user registry on Solaris in a clustered environment**
  Depending on the type of data that is exchanged between IBM WebSphere Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal.

- **Adding a database user registry on Solaris in a clustered environment**
  Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

- **Adding realm support on Solaris in a clustered environment**
  A realm is a group of users from one or more user registries that form a coherent group within IBM WebSphere Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

**Parent topic:** Choosing your user registry model on Solaris in a clustered environment
Configuring a federated LDAP user registry on Solaris in a clustered environment

Depending on the type of data that is exchanged between IBM® WebSphere® Application Server, IBM WebSphere Portal, and your LDAP server, you can either configure your LDAP server over SSL or configure it to have direct access to IBM WebSphere Application Server and IBM WebSphere Portal. Choose one of the following options to configure your LDAP server:

- **Adding an LDAP user registry on Solaris in a clustered environment**
  Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time. If IBM Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=groupName` and the hierarchical format is `cn=groupName, o=root`. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as `wpsadmin`, this ID cannot exist in the LDAP you are adding.

- **Adding an LDAP user registry over SSL on Solaris in a clustered environment**
  Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

**Parent topic:** Configuring the default federated repository on Solaris in a clustered environment
Adding an LDAP user registry on Solaris in a clustered environment

Add an LDAP user registry to the default federated repository to store user account information for authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time. If IBM® Lotus® Domino® will be one of your user registries in a multiple registry configuration and will share a realm with another user registry, ensure that the groups are stored in a hierarchical format in the Domino Directory as opposed to the default flat-naming structure. For example, the flat-naming convention is `cn=groupName` and the hierarchical format is `cn=groupName, o=root`. You must also ensure that the IDs that are unique between the default federated repository and the LDAP you are adding. For example, if the default federated repository contains an ID such as `wpsadmin`, this ID cannot exist in the LDAP you are adding.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

**Tip:** Perform these steps on the primary node only.

**Note:** Use the `wp_add_federated_xxx.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_add_federated_xxx.properties` helper file.

1. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
2. Required: Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated LDAP Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `federated.ldap.id`
   - `federated.ldap.host`
   - `federated.ldap.port`
   - `federated.ldap.bindDN`
   - `federated.ldap.bindPassword`
   - `federated.ldap.ldapServerType`
   - `federated.ldap.baseDN`
3. Required: Enter a value for the following required entity types parameters in the `wkplc.properties` file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - `federated.ldap.et.group.objectClasses`
   - `federated.ldap.et.group.objectClassesForCreate`
   - `federated.ldap.et.group.searchBases`
   - `federated.ldap.et.personaccount.objectClasses`
4. Required: Enter a value for the following required group member parameters in the wkplc.properties file under the Group member attribute heading: 
   Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - federated.ldap.et.personaccount.objectClassesForCreate
   - federated.ldap.et.personaccount.searchBases

5. Save your changes to the wkplc.properties file.

6. Optional: Enter the following Web content authors parameters in the wkplc_comp.properties file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product. Note: See the wkplc_comp.properties file for specific information about the required parameters and for advanced parameters.
   - WcmContentAuthorsGroupId
   - WcmContentAuthorsGroupCN

7. Run the ./ConfigEngine.sh validate-federated-ldap -DWasPassword=password task to validate your LDAP server settings. Attention: If you have not deleted the default file repository, WasPassword is the value entered during installation and not a value found in your LDAP user registry.
   Note: During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press y then Enter.

8. Run the ./ConfigEngine.sh wp-create-ldap -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add an LDAP user registry to the default federated repository. Note: Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

10. Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:
    A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
    B. Enter a value for the following required parameters in the wkplc.properties file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms:
       Note: See the properties file for specific information about the required parameters and for advanced parameters.
       - id
       - baseDN
       - nameInRepository
    C. Save your changes to the wkplc.properties file.
    D. Run the ./ConfigEngine.sh wp-create-base-entry -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to create a base entry in a repository.
    E. Stop and restart all necessary servers to propagate your changes.

11. Optional: Run the ./ConfigEngine.sh wp-query-repository -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to list the names and types of configured repositories.

12. Run the ./ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to check that all defined attributes are available in the configured
LDAP user registry. **Important:** When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

13. Perform the following steps to update the user registry where new users and groups are stored:

   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - `personAccountParent`
      - `groupParent`
      - `personAccountRdnProperties`
      - `groupRdnProperties`

      The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:

      - `personAccountParent=dc=yourco,dc=com`
      - `groupParent=dc=yourco,dc=com`

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to delete the old attributes before adding the new attributes.

   E. Stop and restart all necessary servers to propagate your changes.

14. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for `realmName` or leave blank to update the default realm.

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

   E. Stop and restart all necessary servers to propagate your changes.

15. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

    **Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

    A. Edit the `wp_profile_root` `/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties` file.

    B. Add the following lines to the file:

    ```
    ccn=contentauthors, o=defaultWIMFileBasedRealm -> content_authors_group_DN
    ```
Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `.ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password task`, located in the `wp_profile_root/ConfigEngine` directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured. Table 1. Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone.ldap.realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated.realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated.realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

16. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

17. Optional: **This step is required in a production environment.** Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

   **Important:**
   - Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
   - Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

   **Note:** If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

A. Run the following task from the `wp_profile_root/ConfigEngine` directory: `.ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword`. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` parameter.

B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

C. Run the following task from the `wp_profile_root/ConfigEngine` directory: `.ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid`. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.
18. Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it.

The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: *Deleting the repository.*

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring a federated LDAP user registry on Solaris in a clustered environment

**Related tasks**

- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation

**Related information**

- User IDs and passwords
Adding an LDAP user registry over SSL on Solaris in a clustered environment

Add an LDAP user registry over SSL to the default federated repository to store user account information for secure authorization. You can add multiple LDAP user registries to the default federated repository although you can only add one LDAP server at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add an LDAP user registry over SSL to the default federated repository; you must repeat these steps for each additional LDAP user registry that you plan to add:

**Tip**: Perform these steps on the primary node only.

**Note**: Use the `wp_add_federated_xxx.properties` helper file, located in the `wp_profile_root/ConfigEngine/config/helpers` directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the `wkplc.properties` file, you will use your `wp_add_federated_xxx.properties` helper file.

1. To specify the LDAP server's SSL certificate in the default client trust store, perform either the steps for the server trust store or the client trust store:
   - **Server trust store**
     A. Add the certificate to the trust store:
        1. Log in to the WebSphere Application Server Administrative Console.
        2. Navigate to **Security > SSL certificate and key management > SSL configurations**.
        3. Click the appropriate SSL configuration from the list. For example,
           - **Stand-alone environments**: `NodeDefaultSSLSettings`
           - **Clustered environments**: `CellDefaultSSLSettings`
        
           **Clustered environments**: Ensure the setting for **SSL configuration for outbound connection** matches your SSL settings.
        4. Click **Key stores and certificates**.
        5. Click the appropriate trust store from the list; for example, `NodeDefaultTrustStore`.
        6. Click **Signer certificates**, click **Add**, and then enter the following information:
           - Type the **Alias** the key store uses for the signer certificate.
           - Type the **File name** where the signer certificate is located.
        7. Click **OK** and then click **Save** to save the changes to the master configuration.
   B. Retrieve the certificate from the port:
      1. Log in to the WebSphere Application Server Administrative Console.
2. Navigate to Security > SSL certificate and key management > SSL configurations.
3. Click the appropriate SSL configuration from the list. For example,
   - Stand-alone environments: NodeDefaultSSLSettings
   - Clustered environments: CellDefaultSSLSettings
   **Clustered environments:** Ensure the setting for SSL configuration for outbound connection matches your SSL settings.
4. Click Key stores and certificates.
5. Click the appropriate trust store from the list; for example, NodeDefaultTrustStore.
6. Click Signer certificates, click Retrieve from port, and then enter the following information:
   - Type the Host name used when attempting to retrieve the signer certificate from the SSL port.
   - Type the SSL Port used when attempting to retrieve the signer certificate.
   - Type the Alias the key store uses for the signer certificate.
7. Click Retrieve signer information to retrieve the certificate from the port.
8. Click OK and then click Save to save the changes to the master configuration.
   **Client trust store** Note: This task might report an error, but it does successfully update the trust store. You can ignore the error message.
A. See Secure installation for client signer retrieval.
B. Run the retrieveSigners task from the wp_profile_root/bin directory; see retrieveSigners command for information. In a deployed environment, you will need to run the retrieveSigners task, for any federated node, against the Deployment Manager. Note: This task might report an error, but it does successfully update the trust store. You can ignore the error message.
C. Update the trust store properties file:
   1. Use a text editor to open the ssl.client.props file, located in the wp_profile_root/properties directory.
   2. Change the com.ibm.ssl.trustStore parameter and the related trust store parameters to match the trust file specified in the SSL configuration. For example, enter com.ibm.ssl.trustStore=${CONFIG_ROOT}\cells\wpsbvt\nodes\wpsbvt\trust.p12 to use the default trust store.
   3. Save your changes.
2. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
3. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM Federated LDAP Properties heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - federated.idp.id
   - federated.idp.host
   - federated.idp.port
   - federated.idp.bindDN
   - federated.idp.bindPassword
   - federated.idp.idpServerType
   - federated.idp.baseDN
4. Required: Enter a value for the following required entity types parameters in the wkplc.properties file under the LDAP entity types heading: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.
   - federated.idp.et.group.objectClasses
   - federated.idp.et.group.objectClassesForCreate
   - federated.idp.et.group.searchBases
Required: Enter a value for the following required group member parameters in the wkplc.properties file under the Group member attribute heading:Note: See the properties file for specific information about the required parameters and for advanced parameters.
- federated.idap.gm.groupMemberName
- federated.idap.gm.objectClass
- federated.idap.gm.scope
- federated.idap.gm.dummyMember

Enter a value for the following parameters to enable Secure Socket Layers (SSL):Note: See the properties file for specific information about the required parameters and for advanced parameters.
Required parameters:
- federated.idap.sslEnabled
- federated.idap.sslConfiguration

Optional parameters:
- federated.idap.certificateMapMode
- federated.idap.certificateFilter

Save your changes to the wkplc.properties file.

Optional: Enter the following Web content authors parameters in the wkplc_comp.properties file under the Web Content Management attribute heading and then save your changes. This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates provided with the product.Note: See the wkplc_comp.properties file for specific information about the required parameters and for advanced parameters.
- WcmContentAuthorsGroupId
- WcmContentAuthorsGroupCN

Run the ./ConfigEngine.sh validate-federated-ldap -DWasPassword=password task to validate your LDAP server settings. Attention: If you have not deleted the default file repository, WasPassword is the value entered during installation and not a value found in your LDAP user registry.

Note: During the validation task, you may receive the following prompt: Add signer to the trust store now?. Press y then Enter.

Run the ./ConfigEngine.sh wp-create-ldap -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add an LDAP user registry to the default federated repository. Note: Users who are not in an LDAP do not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

Optional: Perform the following steps to create additional base entries within the LDAP user registry; repeat these steps for each base entry that you want to create for multiple realm support:
A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

B. Enter a value for the following required parameters in the wkplc.properties file under the VMM repository base entry configuration heading to create additional base entries within the LDAP user registry to use when creating realms:

Note: See the properties file for specific information about the required parameters and for advanced parameters.
C. Save your changes to the wkplc.properties file.

D. Run the `./ConfigEngine.sh wp-create-base-entry -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to create a base entry in a repository.

E. Stop and restart all necessary servers to propagate your changes.

13. Optional: Run the `./ConfigEngine.sh wp-query-repository -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to list the names and types of configured repositories.

14. Run the `./ConfigEngine.sh wp-validate-federated-ldap-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to check that all defined attributes are available in the configured LDAP user registry. Important: When you finish configuring your LDAP user registry, see "Adapting the attribute configuration" for information about adding and mapping attributes to ensure proper communication between WebSphere Portal and the LDAP server.

15. Perform the following steps to update the user registry where new users and groups are stored:

   **Note:** If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   **Attention:** During installation, the default file repository creates a default value in the `personAccountRdnProperties` and `groupRdnProperties` parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM supported entity types configuration heading:

      **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

      - `personAccountParent`
      - `groupParent`
      - `personAccountRdnProperties`
      - `groupRdnProperties`

      The parameters `groupParent` and `personAccountParent` must be set to the same value. For example:

      - `personAccountParent=dc=yourco,dc=com`
      - `groupParent=dc=yourco,dc=com`

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to delete the old attributes before adding the new attributes.

   E. Stop and restart all necessary servers to propagate your changes.

16. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

   A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

   B. Enter a value for `realmName` or leave blank to update the default realm.

   C. Save your changes to the `wkplc.properties` file.

   D. Run the `./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.
E. Stop and restart all necessary servers to propagate your changes.

17. Optional: Run the Member Fixer task to update the member names used by Web Content Management with the corresponding members in the LDAP directory. This step ensures that access to the Web content libraries for the Intranet and Internet Site Templates for the contentAuthors group is correctly mapped to the appropriate group in the LDAP directory.

**Note:** This step is only needed if you have installed the product with Web Content Management and intend to use the Intranet and Internet Site Templates that were optionally installed with the product by running the `configure-express` task.

A. Edit the `wp_profile_root`/PortalServer/wcm/shared/app/config/wcmservices/MemberFixerModule.properties file.

B. Add the following lines to the file:

```
uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN

cn=contentauthors,o=defaultWIMFileBasedRealm -> content_authors_group_DN
```

Replace `portal_admin_DN` with the distinguished name of the portal administrator and `content_authors_group_DN` with the distinguished name of the content authors group used during LDAP configuration.

C. Save your changes and close the file.

D. Run the `./ConfigEngine.sh action-express-memberfixer -DmemberfixerRealm=realm_name -DPortalAdminPwd=password -DWasPassword=password` task, located in the `wp_profile_root`/ConfigEngine directory. **Note:** Choose the appropriate value to enter for `realm_name` depending on the type of LDAP user registry you configured. Table 1. **Value for `realm_name` when running the Member Fixer task to update the member names used by Web Content Management**

<table>
<thead>
<tr>
<th>Type of LDAP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>standalone ldap realm</code> in the <code>wkplc.properties</code> file.</td>
</tr>
<tr>
<td>Federated LDAP</td>
<td>The value specified for <code>realm_name</code> should match the value for <code>federated realm</code> in the <code>wkplc.properties</code> file. If the value for <code>federated realm</code> is empty, use <code>defaultWIMFileBasedRealm</code> as the default value.</td>
</tr>
</tbody>
</table>

18. If you have created any additional Web Content Management libraries, run the Web content member fixer task to update the member names used by the libraries.

19. Optional: **This step is required in a production environment.** Before removing the file system repository, perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID and administrative group ID with users and groups that exist in the LDAP user registry:

**Important:**
- Before changing the user ID and password, review Special characters in user ID and passwords located under Planning for WebSphere Portal.
- Ensure the new user ID of the WebSphere Application Server administrator is not identical to the one that you are replacing. Duplicate user IDs cause authentication problems with the administrative console.

**Note:** If you run these tasks after you create your cluster, you must run them on all nodes in the cluster.

A. Run the following task from the `wp_profile_root`/ConfigEngine directory: `./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword` **Important:** You must provide the full distinguished name (DN) for the `newAdminId` parameter.
B. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

C. Run the following task from the `wp_profile_root/ConfigEngine directory`: `./ConfigEngine.sh wp-change-portal-admin-user -DWaspPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid`. **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

**Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-skip.ldap.validation=true` parameter to the task to skip the validation.

D. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

20. Optional: **This step is required in a production environment.** Remove the file system repository if you do not use it.

The federated file system user repository that was the default security setting might not be required after federating the user repository. If the file system repository is no longer needed, removing it can help prevent conflicts created by duplicate user identities existing in multiple repositories. See the following topic, which is organized by operating system, for instructions: *Deleting the repository.*

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring a federated LDAP user registry on Solaris in a clustered environment

**Related tasks**

- [Starting and stopping servers, deployment managers, and node agents](#)
- [Enabling LDAP security after cluster creation](#)

**Related information**

- [User IDs and passwords](#)
Adding a database user registry on Solaris in a clustered environment

Add a database user registry to the default federated repository to store user account information for authentication and authorization. You can add multiple database user registries to the default federated repository although you can only add one database user registry at a time.

In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

If you have WebSphere Application Server Version 7.0.x installed, you must install APAR PM23090 and APAR PM24181 for WebSphere Portal prior to running this task. Clusters with WebSphere Application Server Version 6.1.x do not require these APARs.

Perform the following steps to add a database user registry to the default federated repository; you must repeat these steps for each additional database user registry that you plan to add:

Tip: Perform these steps on the primary node only.

Note: Use the wp_add_DB.properties helper file, located in the wp_profile_root/ConfigEngine/config/helpers directory, when performing this task to ensure the correct properties are entered. In the instructions below, when the step refers to the wkplc.properties file, you will use your wp_add_DB.properties helper file.

1. Set up a new database, including creating a new user with appropriate database privileges for accessing the database:

   Instructions for setting up databases: Refer to the appropriate documentation for the type of database you want to set up.

   Consulting your database administrator: The task of setting up a new database is typically performed by a database administrator. However, the following steps are provided for your reference in the event you are creating a stand-alone database for testing or demonstration purposes. Consult your database administrator before proceeding with the following steps if you plan to create a database for a production environment.

   Table 1. Steps for creating a new database to use as a database user registry.
<table>
<thead>
<tr>
<th>DB2</th>
<th>Perform the following steps to create a DB2 database: Install DB2. Enter the following database tuning commands:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Oracle</th>
<th>Perform the following steps to create an Oracle database: Install Oracle using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16. Configure the database in Dedicated Server Mode. Enter the recommended initial buffer pool sizes or set them according to your business needs:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>db_block_size = 8192</td>
</tr>
<tr>
<td></td>
<td>db_cache_size = 800M</td>
</tr>
<tr>
<td></td>
<td>db_files = 1024</td>
</tr>
<tr>
<td></td>
<td>log_buffer = 65536</td>
</tr>
<tr>
<td></td>
<td>open Cursors = 1500</td>
</tr>
<tr>
<td></td>
<td>pga_aggregate_target = 200M</td>
</tr>
<tr>
<td></td>
<td>pre_page_sga = true</td>
</tr>
<tr>
<td></td>
<td>processes = 300</td>
</tr>
<tr>
<td></td>
<td>shared_pool_size = 200M</td>
</tr>
</tbody>
</table>

| SQL Server | Perform the following steps to create an SQL Server database: Install SQL Server. Set Collation to case-sensitive. Note: Install SQL Server with the appropriate portal database collation so that your tempdb collation setting matches the collation you use for the property extension database. The tempdb collation is inherited from the master database, which you set when you install SQL Server. |

2. Perform the following steps to define the **DbDriver** and **DbLibrary** parameter values:
   A. Navigate to the following directory: `wp_profile_root/ConfigEngine/properties`
   B. Locate and open `wpklc_dctype.properties` with any text editor.
   C. Enter a value for the following parameters under the appropriate database type properties heading:
      - `db_type.DbDriver`
      - `db_type.DbLibrary`
   D. Save your changes.

3. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

4. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Federated Database Properties heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.
5. Change the value for the `com.ibm.SOAP.requestTimeout` parameter to 1000.
   A. Navigate to the following directory: `wp_profile_root/properties`
   B. Locate and open `soap.client.props` with any text editor.
   C. Locate the `com.ibm.SOAP.requestTimeout` parameter and change the value to 1000.
   D. Save and close `soap.client.props`.

6. Perform the following steps in a clustered environment:
   A. Run the `./ConfigEngine.sh wp-prep-vmm-db-secured-environment -DWasPassword=password -
      DDbDomain=federated.db -Db_type.DmgrDbLibrary=local path of the database jars on the Deployment
      Manager -DDmgrNodeName=dmgr_node_name` task from the `wp_profile_root/ConfigEngine` directory to create the local
      Deployment Manager WebSphere variable used to access the database jars. **Note:** The `db_type` in `db_type`
      `.DmgrDbLibrary` should be set to the type of database you are using, for example `db2`. The `local full path of the`
      `database jars on the Deployment Manager` should be one of the following options:
      - `DB2 Type 2 driver:db2java.zip`
      - `DB2 Type 4 driver:db2jcc.jar;db2jcc_license_cu.jar`
      - `DB2 for z/OS Type 2 driver:db2java.zip`
      - `DB2 for z/OS Type 4 driver:db2jcc.jar;db2jcc_license_cisuz.jar;db2jcc javax.jar`
      - `Oracle:ojdbc14.jar`
      - `SQL Server JDBC driver provided by Microsoft:sqljdbc.jar`
      - `SQL Server JDBC driver provided by DataDirect:sqlserver.jar;base.jar;util.jar`
   B. Run the following task. Include each node name as a comma separated list in the command:
      **Running the task:** You do not have to run this task more than once. You can run this task from any node in the
      cluster.
      1. Set the property value for `federated.db.DbType` in the `wkplc.properties` file if using a database user registry.
      2. Run the `./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=password -
         DDbDomain=federated.db -DVmmNodeName=node_name -Db_type.NodeDbLibrary=local full path of the
         database jars` task from the `wp_profile_root/ConfigEngine` directory on each node to create the variable used to
         access the VMM database jars. **Note:** The `VmmNodeName` is a list of one or more WebSphere Portal nodes names in
         the cell which share the same database driver paths. The `db_type` in `db_type.NodeDbLibrary` should be set to
         the type of database you are using, for example `db2`.
      C. Stop and restart all necessary servers to propagate your changes.
   7. Run the `./ConfigEngine.sh wp-create-db -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine`
      directory, to add a database user registry to the default federated repository. **Note:** Users who are not in an LDAP do
not have awareness and cannot see if other users are online. This can happen if you install WebSphere Portal and then enable a Federated LDAP or Federated database user repository that does not contain that user. Also, users who sign up using the Self Care portlet do not have awareness.

8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

9. Perform the following steps to update the user registry where new users and groups are stored:

   Note: If you are using multiple LDAP user registries and/or a database user registry, only run this task for the user registry that you want to define as the default user registry where new users and groups are stored.

   Attention: During installation, the default file repository creates a default value in the personAccountRdnProperties and groupRdnProperties parameters. To change the default value, you must run this task twice; once to clear the default value and once to add the new value.

A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

B. Enter a value for the following required parameters in the wkplc.properties file under the VMM supported entity types configuration heading:

   Note: See the properties file for specific information about the required parameters and for advanced parameters.
   - personAccountParent
   - groupParent
   - personAccountRdnProperties
   - groupRdnProperties
   The parameters groupParent and personAccountParent must be set to the same value. For example:
   - personAccountParent=dc=yourco,dc=com
   - groupParent=dc=yourco,dc=com

C. Save your changes to the wkplc.properties file.

D. Run the ./ConfigEngine.sh wp-set-entitytypes -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to delete the old attributes before adding the new attributes.

E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Run the ./ConfigEngine.sh wp-query-repository -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to list the names and types of configured repositories.

    If you created your clustered environment then performed the steps in this task, you must now run the enable-jcr-security task on the secondary node. See Enabling LDAP security after cluster creation for instructions.

Parent topic: Configuring the default federated repository on Solaris in a clustered environment

Related tasks
Starting and stopping servers, deployment managers, and node agents
Enabling LDAP security after cluster creation
Adding realm support on Solaris in a clustered environment

A realm is a group of users from one or more user registries that form a coherent group within IBM® WebSphere® Portal. Realms allow flexible user management with various configuration options. A realm must be mapped to a Virtual Portal to allow the defined users to log in to the Virtual Portal. When configuring realm support, you can perform these steps for each base entry that exists in your LDAP and/or database user registry to create multiple realm support.

Before configuring realm support, you must add all LDAP user registries and/or database user registries, that you will use to create a single realm or multiple realms, to the federated repository. If you are going to create multiple realms, you must create all required base entries within your LDAP user registries and/or database user registries. All base entry names must be unique within the federated repository. In single server environments, you do not have to start or stop the WebSphere_Portal and server1 servers to complete the following steps. In clustered environments, you must stop all application servers on your system, including WebSphere_Portal, then start the nodeagent and deployment manager servers before you begin any of the following steps.

Perform the following steps to add realm support to your user registry model:

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.

2. Required: Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading:

   - realmName
   - securityUse
   - delimiter
   - addBaseEntry

   Note: See the properties file for specific information about the required parameters and for advanced parameters.

3. Save your changes to the wkplc.properties file.

4. Run the ./ConfigEngine.sh wp-create-realm -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add a new realm to the Virtual Member Manager configuration. Important: To create multiple realms, ensure that your federated repository contains the required unique base entries. Stop and restart the appropriate servers for your installation environment, and then update the wkplc.properties file with the base entry information and rerun the wp-create-realm task. Repeat these steps until all realms are created.

5. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

6. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading and then save your changes:

   - realmName
   - realm.personAccountParent
   - realm.groupParent
   - realm.orgContainerParent

   Note: See the properties file for specific information about the required parameters and for advanced parameters.

7. Run the ./ConfigEngine.sh wp-modify-realm-defaultparents -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to update the default parents per entity type and realm. Important: Stop and restart the appropriate servers for your installation environment before rerunning this task for any additional entity types.
and realms.

8. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

9. Optional: Perform the following steps to add additional base entries to the realm configuration; for example, if you had two additional base entries (base entry 1 and base entry 2) to add to the realm you just created, you would update the wkplc.properties file with the information from base entry 1 and then run this task. Then you would update the properties file with the information for base entry 2 and then run this task:
   A. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
   B. Enter a value for the following required parameters in the wkplc.properties file under the VMM realm configuration heading:
      - realmName
      - addBaseEntry
   C. Save your changes to the wkplc.properties file.
   D. Run the ./ConfigEngine.sh wp-add-realm-baseentry -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, to add additional LDAP base entries to the realm configuration.
   E. Stop and restart all necessary servers to propagate your changes.

10. Optional: Perform the following steps to replace the WebSphere Application Server and WebSphere Portal administrator user ID; this step is required if you change the default realm:
    A. Create a new user in the Manage Users and Groups portlet to replace the current WebSphere Application Server administrative user.
    B. Create a new user in the Manage Users and Groups portlet to replace the current WebSphere Portal administrative user.
    C. Create a new group in the Manage Users and Groups portlet to replace the current group.
    D. Run the ./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=password -DnewAdminId=newadminId -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupId task, from the wp_profile_root/ConfigEngine directory, to replace the old WebSphere Application Server administrative user ID and group ID with the new user and group.Important: You must provide the full distinguished name (DN) for the newAdminId and newAdminGroupId parameters.
    E. Additional parameter for stopped servers: This task verifies the user against a running server instance. If the server is stopped, add the -Dskip.ldap.validation=true parameter to the task to skip the validation.
    F. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.
    G. Run the ./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminId -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupId task to replace the old WebSphere Portal administrative user ID and group ID with the new user and group.Important: You must provide the full distinguished name (DN) for the newAdminId and newAdminGroupId parameters.
    H. Additional parameter for stopped servers: This task verifies the user against a running server instance. If the server is stopped, add the -Dskip.ldap.validation=true parameter to the task to skip the validation.
    I. Verify that the task completed successfully. In a clustered environment, restart the deployment manager, the node agent(s), and WebSphere_Portal servers. In a standalone environment, restart the server1 and WebSphere_Portal servers.

11. Optional: Perform the following steps to set the realm you created as the default realm:Remember: Only users defined in base entries that exist in the default realm are able to log into WebSphere Portal. If you find that a user cannot log in to WebSphere Portal, check to see if the base entry that contains the user exists in the default realm. You can run the wp-query-realm-baseentry task to see what base entries are part of the default realm. If the default realm is missing
the base entry, run the `wp-add-realm-baseentry` task to add the base entry to the default realm.

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. For `defaultRealmName`, type the `realmName` property value you want to use as the default realm.

C. Save your changes to the `wkplc.properties` file.

D. Run the `./ConfigEngine.sh wp-default-realm -DWasPassword=password` task, from the `wp_profile_root` directory, to set this realm as the default realm.

E. Stop and restart all necessary servers to propagate your changes.

12. Optional: Perform the following steps to query a realm for a list of its base entries:

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. For `realmName`, type the name of the realm you want to query.

C. Save your changes to the `wkplc.properties` file.

D. Run the `./ConfigEngine.sh wp-query-realm-baseentry -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, to list the base entries for a specific realm.

13. Optional: Perform the following steps to enable the full distinguished name login if the short names are not unique for the realm:

A. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

B. Enter a value for `realmName` or leave blank to update the default realm.

C. Save your changes to the `wkplc.properties` file.

D. Run the `./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=password` task, located in the `wp_profile_root/ConfigEngine` directory, to enable the distinguished name login. **Note:** After running this task to enable the full distinguished name login, you can run the `./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=password` task to disable the feature.

E. Stop and restart all necessary servers to propagate your changes.

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

**Parent topic:** Configuring the default federated repository on Solaris in a clustered environment

**Related tasks**

- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation
Adapting the attribute configuration

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you will need to adapt the attribute configuration to match the configured LDAP server(s) and your business needs. However, you do not need to perform these steps if you are using either a database user registry or the default federated file-based repository for out-of-box installations.

After installation, IBM WebSphere Portal has a predefined set of attributes for users and groups. Your LDAP server may have a different set of predefined user and group attributes. To ensure proper communication between WebSphere Portal and your LDAP server, you can configure additional attributes and flag existing attributes as required or unsupported on a per repository basis or for all configure repositories.

LDAP servers can only handle attributes that are explicitly defined in their schema. The LDAP server’s schema is different from the WebSphere Portal schema but the two schemas should match for proper communication between WebSphere Portal and the LDAP server. The task to add the LDAP user registry does some basic attribute configurations depending on the type of LDAP server that you choose. You may, however, still need to adapt the WebSphere Portal configuration to match the LDAP schema; for example, if an attribute is defined in WebSphere Portal but not in the LDAP server, you will need to perform one of the following tasks to resolve this mismatch:

- Flag the attribute as unsupported for the LDAP server
- Introduce an attribute mapping that maps the WebSphere Portal attribute to an attribute defined in the LDAP schema

Perform the following tasks to adapt the attribute configuration to match the configured LDAP server(s) and your business needs:

1. **Querying the defined attributes on Solaris**
   After installing IBM WebSphere Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

2. **Adding attributes on Solaris in a clustered environment**
   The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM WebSphere Portal and your user registry.

3. **Mapping attributes on Solaris in a clustered environment**
   After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

4. **Removing attributes**
   Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

**Parent topic:** Choosing your user registry model on Solaris in a clustered environment

**Related tasks**
- Adding an LDAP user registry on Solaris
- Adding an LDAP user registry over SSL on Solaris
- Configuring a stand-alone LDAP user registry on Solaris
Configuring a stand-alone LDAP user registry over SSL on Solaris
Querying the defined attributes on Solaris

After installing IBM® WebSphere® Portal and configuring your LDAP user registries, you can query the defined attributes to see what attributes are flagged as unsupported or if the attribute is mapped to a different LDAP attribute.

**Warning:** If you are using a database user registry or a property extension database, copy the database drivers to the `AppServer_root/lib` directory before executing this script.

Run the `./ConfigEngine.sh wp-query-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory, any time during the configuration process or at runtime to query an overview of the currently defined attributes. This task creates the `availableAttributes.html` report, located in the `wp_profile_root/ConfigEngine/log` directory. The report contains one table that lists the available attributes for Users (PersonAccount) and one table that lists the available attributes for Groups. For each configured repository there is a column that indicates if the attribute is flagged as unsupported or if the attribute is mapped to a different LDAP attribute. **Note:** This task does not validate the existence of attributes in the LDAP schema.

**Parent topic:** Adapting the attribute configuration

**Next topic:** Adding attributes on Solaris in a clustered environment
Adding attributes on Solaris in a clustered environment

The VMM is configured with a default attribute schema that might not be compatible with your LDAP server. If this is the case, extend the VMM attribute schema by adding new attributes that you can map between IBM® WebSphere® Portal and your user registry.

Perform the following steps to add new attributes to your user registry:

1. Install the required Enterprise Archive (.ear) file on WebSphere Application Server. Select the appropriate task to install the .ear file:

   **Table 1. Steps for installing the .ear file by environment**

<table>
<thead>
<tr>
<th>Environment</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stand-alone environment</strong></td>
<td>Perform the following steps to install the .ear file in a stand-alone environment: Open a command prompt. Navigate to the <code>wp_profile_root/ConfigEngine</code> directory. Run the <code>./ConfigEngine.sh wp-la-install-ear -DWasPassword=password</code> task.</td>
</tr>
<tr>
<td><strong>Clustered environment</strong></td>
<td>Perform the following steps to install the .ear file in a clustered environment: Open a command prompt. Navigate to the <code>wp_profile_root/ConfigEngine</code> directory on the primary node. Run the <code>./ConfigEngine.sh wp-la-install-ear -DWasPassword=dmgr_password -DServerName=dmgr -DNodeName=node_name</code> task. Where <code>node_name</code> is the name of the node where the deployment manager resides; you can find the <code>node_name</code> value in the WebSphere Application Server Administrative Console under System administrator &gt; Deployment Manager &gt; Runtime tab &gt; General Properties &gt; Node Name.</td>
</tr>
</tbody>
</table>

2. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: Starting and stopping servers, deployment managers, and node agents.

3. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.

4. Enter a value for the following required parameters in the `wkplc.properties` file under the VMM Property Extension Properties heading: Note: See the properties file for specific information about the required parameters and for advanced parameters.

   - `la.providerURL`
   - `la.propertyName`
   - `la.entityTypes`
   - `la.dataType`
   - `la.multiValued`

5. Save your changes to the `wkplc.properties` file.
6. Run the `./ConfigEngine.sh wp-add-property -DWasPassword=password` task to add the attribute to the user registry.

   **Note:** This task performs an EJB call to WebSphere Application Server, which must authenticate against WebSphere Application Server. Depending on the configuration in the `sas.client.props` file, you may receive a popup window or a command line prompt asking for user identity and password. Enter the WebSphere Application Server user ID and password.

   **Remember:** If you have multiple properties to add, repeat all steps, except for the `wp-la-install-ear` task, until all new attributes are added.

7. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents.*

If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.

---

**Parent topic:** Adapting the attribute configuration

**Previous topic:** Querying the defined attributes on Solaris

**Next topic:** Mapping attributes on Solaris in a clustered environment

### Related tasks
- Starting and stopping servers, deployment managers, and node agents
- Enabling LDAP security after cluster creation
Mapping attributes on Solaris in a clustered environment

After you install and configure your LDAP user registry and after you query the defined attributes, you can map the attributes so they match the configured LDAP servers and your business needs.

Perform the following steps to map attributes between WebSphere Portal and your LDAP server; if you have multiple LDAP servers, you will need to perform these steps for each LDAP server:

1. Use a text editor to open the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory.
2. Enter a value for one of the following sets of parameters in the wkplc.properties file to identify your LDAP server:
   - **Note:** Make sure you use the same values you used to configure your LDAP server.
   - **Table 1. Identifying your LDAP server in the wkplc.properties file.**

3. Run one of the following tasks to check that all defined attributes are available in the configured LDAP user registry:
   - **Table 2. Task to check that all defined attributes are available in the configured LDAP user registry.**

4. Open the ConfigTrace.log file, located in the wp_profile_root/ConfigEngine/log directory, to review the following output for the PersonAccount and Group entity type:
- The following attributes are defined in WebSphere Portal but not in the LDAP server
  - This list contains all attributes that are defined in WebSphere Portal but not available in the LDAP. Flag attributes that you do not plan to use in WebSphere Portal as unsupported. Map the attributes that you plan to use to the attributes that exist in the LDAP; you must also map the `uid`, `cn`, `firstName`, `sn`, `preferredLanguage`, and `ibm-primaryEmail` attributes if they are contained in the list.

- The following attributes are flagged as required in the LDAP server but not in WebSphere Portal
  - This list contains all attributes that are defined as "MUST" in the LDAP server but not as required in WebSphere Portal. You should flag these attributes as required within WebSphere Portal; see the step below about flagging an attribute as either unsupported or required.

- The following attributes have a different type in WebSphere Portal and in the LDAP server
  - This list contains all attributes that WebSphere Portal might ignore because the data type within WebSphere Portal and within the LDAP server do not match.

5. Use a text editor to open the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine/properties` directory.
6. Enter a value for one of the following sets of parameters in the `wkplc.properties` file to correct any issues found in the config trace file: `Table 3. Parameters to define in the wkplc.properties file to correct any issues found in the config trace file`.

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td>The following parameters are found under the LDAP attribute configuration heading: <strong>Note</strong>: See the properties file for specific information about the required parameters and for advanced parameters.</td>
</tr>
</tbody>
</table>
|                 | `standalone.id`standalone.ldap.attributes.nonSupportedstandalone.ldap.attributes.nonSupported.deletestandalone.ldap.attributes.mapping.ldapNamestandalone.ldap.attributes.mapping.porth Na
testandalone.ldap.attributes.mapping.entityTypes` For example, the following values will flag `certificate` and `members` as unsupported attributes and will map `ibm-primaryEmail` to `mail` and `ibm-jobTitle` to `title` for both the `PersonAccount` and `Group` entityTypes.  |
|                 | `standalone.id`standalone.ldap.attributes.nonSupportedCertificate=ibm-
|                 | `standalone.id`standalone.ldap.attributes.nonSupported.members=
|                 | `standalone.id`standalone.ldap.attributes.mapping.portarLayout=
|                 | `standalone.id`standalone.ldap.attributes.mapping.entityTypes=
|                 | `PersonAccount, Group` |
7. Save your changes to the `wkplc.properties` file.

8. Run one of the following tasks to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between WebSphere Portal and the LDAP user registry:

   **Table 4. Task to update the LDAP user registry configuration with the list of unsupported attributes and the proper mapping between Portal and the LDAP user registry.**

<table>
<thead>
<tr>
<th>Repository type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone</td>
<td><code>./ConfigEngine.sh wp-update-standalone-ldap-attribute-config -DWasPassword=password</code> task, from the <code>wp_profile_root/ConfigEngine</code> directory</td>
</tr>
<tr>
<td>Federated</td>
<td><code>./ConfigEngine.sh wp-update-federated-ldap-attribute-config -DWasPassword=password</code> task, from the <code>wp_profile_root/ConfigEngine</code> directory</td>
</tr>
</tbody>
</table>

9. Stop and restart the appropriate servers to propagate the changes. For specific instructions, see the following link under Related tasks: *Starting and stopping servers, deployment managers, and node agents.*

10. Optional: Perform the following steps to flag an attribute as either unsupported or required for the entire WebSphere Portal environment instead of just for the specified LDAP:

    A. Enter a value for the following required parameters in the `wkplc.properties` file: **Note:** See the properties file for specific information about the required parameters and for advanced parameters.

        - `user.attributes.required`
        - `user.attributes.nonsupported`

    B. Save your changes to the `wkplc.properties` file.

    C. Run the `./ConfigEngine.sh wp-update-attribute-config -DWasPassword=password` task, from the `wp_profile_root/ConfigEngine` directory.

    D. Stop and restart all necessary servers to propagate your changes.

    If you created your clustered environment then performed the steps in this task, you must now run the `enable-jcr-security` task on the secondary node. See *Enabling LDAP security after cluster creation* for instructions.
Removing attributes

Due to a Virtual Member Manager (VMM) limitation, there is currently no task to update an attribute. Therefore, if you added an attribute to your property extension database or when adapting attributes to match your LDAP server that were spelled incorrectly or already added due to migration, you must remove the attribute from the database. Use caution when performing these steps.

Perform the following steps to remove an attribute from your database:

**Important:** Do not remove attributes that have already been populated with user values because this can cause database inconsistencies.

**Cluster note:** In a clustered environment, perform the following steps on the deployment manager and then resynch the nodes.

1. Perform the following steps to remove an attribute stored in a property extension database; this is an attribute that was added using the `wp-add-la-property` task.
   - A. Open the tool you use to edit your database.
   - B. Verify that your attribute name is available in the LAPROP table.
   - C. Delete the required attributes from the LAPROP table.
   - D. Open the `wimxmlextension.xml` file, located in the `wp_profile_root/config/cells/cellname/wim/model` directory.
   - E. Locate and delete the `propertySchema` definition for the attributes that you deleted from the LAPROP table; for example:
     ```xml
     <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String" multiValued="true" propertyName="attribute_name">
     <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
     </wim:propertySchema>
     ```
   - F. Save your changes to the `wimxmlextension.xml` file.
   - G. Open the `wimconfig.xml` file, located in the `wp_profile_root/config/cells/cellname/wim/config` directory.
   - H. Locate and delete the `propertiesNotSupported` definitions for the attributes that you deleted from the LAPROP table; for example:
     ```xml
     <config:propertiesNotSupported name="attribute_name"/>
     ```
   - I. Save your changes to the `wimconfig.xml` file.
   - J. Stop and restart the server1 and WebSphere_Portal servers from the `wp_profile_root/bin` directory.

2. Perform the following steps to remove an attribute that is not stored in a property extension database:
   - A. Open the `wimxmlextension.xml` file.
   - B. Locate and delete the `propertySchema` definition for the attributes you previously added; for example:
     ```xml
     <wim:propertySchema nsURI="http://www.ibm.com/websphere/wim" dataType="String" multiValued="true" propertyName="attribute_name">
     <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
     </wim:propertySchema>
     ```
   - C. Save your changes to the `wimxmlextension.xml` file.
   - D. Open the `wimconfig.xml` file.
   - E. Locate and delete the stanza that corresponds to the custom attribute you deleted from the `wimextension.xml` file; for example:
     ```xml
     <config:attributes name="attribute_name" propertyName="property_name">
     <config:entityTypes>PersonAccount</config:entityTypes>
     </config:attributes>
     ```
   - F. Save your changes to the `wimconfig.xml` file.
G. Stop and restart the server1 and WebSphere_Portal servers.

**Parent topic:** Adapting the attribute configuration  
**Previous topic:** Mapping attributes on Solaris in a clustered environment
Preparing additional nodes on Solaris

After installing and configuring your primary node, you can create your secondary nodes. You must install IBM® WebSphere® Portal on each node and then configure the node to access the database and user registry before adding it to the cluster.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on Solaris
  - Preparing your Solaris operating system
  - Preparing the primary node on Solaris
  - Installing a secondary node after migrating from previous version

Perform the following tasks to prepare your secondary node:

- **Installing WebSphere Portal on Solaris on the additional nodes**
  Install IBM WebSphere Portal on your secondary nodes to create a highly available and scalable environment.

- **Choosing the type of additional node to create on Solaris**
  If you created a static cluster using the IBM WebSphere Application Server Network Deployment, add additional static nodes to the cluster. If you created a dynamic cluster using IBM WebSphere Virtual Enterprise, add additional dynamic nodes to the cluster.

- **Choosing the type of vertical cluster to create on Solaris**
  If you are using the IBM WebSphere Application Server Network Deployment, you can add vertical cluster members to your static cluster. If you are using IBM WebSphere Virtual Enterprise, you can add vertical cluster members to your dynamic cluster.

**Parent topic:** Setting up a cluster on Solaris

**Previous topic:** Configuring WebSphere Portal to use a user registry on Solaris in a clustered environment

**Next topic:** Tune your servers
Installing WebSphere Portal on Solaris on the additional nodes

Install IBM® WebSphere® Portal on your secondary nodes to create a highly available and scalable environment. Review the Supported hardware and software requirements for your operating system and ensure that all hardware and software matches the minimum product level before installing WebSphere Portal. Review Installation methods, options, and sources before installing to determine your installation source location and installation method. **Note:** If you plan to use IBM WebSphere Application Server Version 7.0, you must purchase and install it before installing WebSphere Portal on the existing WebSphere Application Server Version 7.0 instance.

Perform the following steps to install WebSphere Portal:

1. Type `ping yourserver.yourcompany.com` on a command line to verify that your fully qualified host name is properly configured.
2. If you are installing with an existing WebSphere Application Server instance, ensure it is installed at the supported level and has the following features installed:
   
   **Restriction:** Installing WebSphere Portal Version 6.1.0.1 or higher on an existing WebSphere Application Server Version 7.0 instance where WebSphere Portal Version 7.0 was installed is unsupported.

   **Important:** Besides ensuring that the existing WebSphere Application Server instance is at the supported level, you will also need to ensure that Apache Derby is installed at the supported level before installing WebSphere Portal. See WebSphere Portal detailed system requirements for supported levels.

   - Application Server
   - Administration
     - Scripted Administration
     - Administrative Console
   - Ant and Deployment Tools
     - Deploy Tool
     - Ant Utilities

3. Optional: Perform the following steps if you want to install WebSphere Portal using a non-root user:

   **Restriction:** Although it is possible to install WebSphere Application Server as a non-root user, there are limitations to this option that can affect WebSphere Portal functionality, which is why you should install WebSphere Application Server as a root user.

   **A.** Install the current supported version of WebSphere Application Server as a root user.

   **B.** Open a command prompt and perform the following steps to create a non-root user and to change ownership:

   1. Use the appropriate system commands to create a new group, to create a new user, and to add the new user to the new group.
      
      `chgrp -R group_name/opt/IBM`
      
      `chmod -R g+rw /opt/IBM`
      
      `chgrp -R group_name /tmp`
      
      `chmod -R g+wr /tmp`
      
      `chgrp -R group_name /var/tmp`
      
      `chmod -R g+wr /var/tmp`
chgrp -R group_name /var/tmp

C. Launch the installation program per the next step. During the installation, select the existing WebSphere Application Server using the non-root user and set the installation location to a unique location under the path where non-root permissions were granted.

4. Choose one of the following installation commands based on the installation method you want to use to install WebSphere Portal; see Installation Methods for more information: Advance installation parameters: To customize your installation, you can add various parameters to your installation commands; for example, you can specify your port information. See Advanced installation parameters for information.

**Restriction:** When defining your installation path, the **ONLY** supported characters are ASCII letters [A-Z and a-z], numbers [0-9], slashes [/ or \, depending on your operating system], and the dash [-].

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphical user interface</td>
<td>./install.sh</td>
</tr>
<tr>
<td>Console mode</td>
<td>./install.sh -console from the PortalExpress subdirectory</td>
</tr>
<tr>
<td>Silent Install</td>
<td>./install.sh -options &quot;path_to_file/response_filename&quot;, where path_to_file is the full path to the response file and response_filename is the name of the file. A sample install response file (installresponse.txt) and a sample uninstall response file (uninstallresponse.txt) are located in the setup CD root directory. <strong>Important:</strong> Do not place the response file in a path that contains a space and do not put a space in the file name.</td>
</tr>
</tbody>
</table>

**Important:** If you have a Solaris 9 on Sun SPARC, you MUST either add the `-W defaults.force32bit=true` parameter to your installation command or install the 32-bit WebSphere Application Server product before installing WebSphere Portal to prevent the installer from automatically selecting a 64-bit environment.

5. Verify your installation was successful; access WebSphere Portal using the `http://yourserver:yourport/wps/portal` format. Run the `./ConfigEngine.sh list-server-ports -DWasPassword=password` task from the `wp_profile_root/ConfigEngine/log/wp_PortMatrix.txt` file that lists WebSphere Portal ports for your installation.

6. Optional: Perform the following steps if you need to change the ports for WebSphere Application Server or WebSphere Portal:

**Note:** The starting port parameter is required for a successful completion of the `modify-ports-by-startport` task. Once you specify a start port, this port becomes the base for assigning port values. The code increments this value as each port is assigned, which means that the WebSphere Portal ports will be assigned incrementally starting with the port defined with the `modify-ports-by-startport` task.

A. Stop the server1 and WebSphere Portal servers.
B. Run one of the following commands for each server you need to change:

<table>
<thead>
<tr>
<th>Method</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting port number</td>
<td>./ConfigEngine.sh modify-ports-by-startport -DWasPassword=password -DModifyPortsServer=servername -DStartPort=starting port number</td>
</tr>
</tbody>
</table>
C. Restart the server1 and WebSphere_Portal servers.

7. To prevent Out of Memory errors, perform the following steps to set the MaxPermSize:

   Tip: If Maximum Heap Size is set to 2048 M or higher, set the MaxPermSize to a quarter of the value entered for the Maximum Heap Size; for example, if your Maximum Heap Size is 3000 M, set MaxPermSize to 750 M. If your Maximum Heap Size is less than 2048 M, set MaxPermSize to 512 M.

   A. Log in to the WebSphere Application Server Administration Console.
   B. Click Servers > Server Types > WebSphere application servers > WebSphere_Portal.
   D. In the Generic JVM arguments field, change the MaxPermSize value to -XX:MaxPermSize=numeric value, where numeric value is a quarter of the value entered for the Maximum Heap Size. Important: If MaxPermSize does not exist in the Generic JVM arguments field, add it to the field but do not replace existing information in the Generic JVM arguments field with the MaxPermSize information.
   E. Click OK to save your changes.
   F. Click Save to save your changes to the master configuration.
   G. Log out of the WebSphere Application Server Administration Console.
   H. Restart the server1 and WebSphere_Portal servers.

8. Perform the following steps if you migrated from a previous version of WebSphere Portal:

   A. Log on to the WebSphere Application Server Administrative Console.
   C. Click WP Identification.
   D. Click Custom properties.
   E. If applicable, note the location of the LpidToGupidMapping.properties file.
   F. If the LpidToGupidMapping.properties file exists on the primary node, copy the file to the same location on the secondary node.

Parent topic: Preparing additional nodes on Solaris

Related concepts

Installation methods
IBM Support Assistant Lite for WebSphere Portal

Note: Sample port files are available on the Setup disc.
Related reference
Advanced installation parameters
Choosing the type of additional node to create on Solaris

If you created a static cluster using the IBM® WebSphere® Application Server Network Deployment, add additional static nodes to the cluster. If you created a dynamic cluster using IBM WebSphere Virtual Enterprise, add additional dynamic nodes to the cluster.

Choose one of the following options to add additional nodes to your cluster:

- **Adding additional nodes to the static cluster on Solaris**
  After installing IBM WebSphere Portal on your secondary node, you must add the node to the cluster to create a highly available environment.

- **Adding an additional node to an existing dynamic cluster**
  After creating your dynamic cluster, you can add additional nodes to expand the capacity of the dynamic cluster. Install and configure the additional node, then add it to the existing dynamic cluster node group, and then add it to the existing IBM WebSphere Virtual Enterprise dynamic cluster.

**Parent topic:** Preparing additional nodes on Solaris
Adding additional nodes to the static cluster on Solaris

After installing IBM® WebSphere® Portal on your secondary node, you must add the node to the cluster to create a highly available environment.

Perform the following steps to add the secondary node to the cluster:

1. Perform the following steps on the secondary node to access your database server:
   Note: All properties files are located in the `wp_profile_root`/ConfigEngine/properties directory.
   A. For JDBC Type 2 connections only, ensure that the database client software is installed on the secondary node using the same settings as the primary node and that you can connect to the remote database. It is not necessary to create databases or users when configuring secondary nodes. **DB2 users:** When using DB2, you must catalog the TCP/IP node and all databases used on each secondary node. For more information refer to Preparing DB2.
   B. For JDBC Type 4 connections only, copy the JDBC Type 4 jar files to the secondary node from the DB2 server.
   C. Set all database properties in the `wkplc_dbtype.properties` and `wkplc_comp.properties` files with the appropriate values from the primary node.

2. Run the following tasks to validate your database settings:
   A. `./ConfigEngine.sh validate-database-driver` -
      `DTransferDomainList=release,customization,community,jcr,feedback,likeminds` `-DWasPassword=password`
   B. `./ConfigEngine.sh validate-database-connection` `-DWasPassword=password` -
      `DTransferDomainList=release,customization,community,jcr,feedback,likeminds`

3. Stop the server1 and WebSphere_Portal servers on the secondary node and ensure that the following parameters are set correctly in the `wkplc.properties` file, located in the `wp_profile_root`/ConfigEngine/properties directory of the secondary node:
   Note: Although you can add these parameters directly to any task that you run while creating your cluster, you may want to add them to the properties file while creating your cluster and then remove them when you are finished to keep your environment secure.
   A. Verify that `WasUserid` is set to your deployment manager administrator ID.
   B. Verify that `WasPassword` is set to your deployment manager administrator password.
   C. Verify that `PortalAdminId` is set to your WebSphere Portal administrator ID.
   D. Verify that `PortalAdminPwd` is set to your WebSphere Portal password.
   E. Verify that `WasRemoteHostName` is set to the fully qualified host name of the deployment manager.
   F. Verify that `WasSoapPort` is set to the SOAP port that the deployment manager is using; the default value is 8879.
   G. Verify that `PrimaryNode` is set to `false`.
   H. Verify that `ClusterName` is set to the primary node’s `ClusterName`.

4. Run the `./ConfigEngine.sh cluster-node-config-pre-federation` `-DWasPassword=dmgr_password` task, from the `wp_profile_root`/ConfigEngine directory of the secondary node.

5. Optional: If you configured a database user registry or a property extension (lookaside) database, run the following task to set up access to the database drivers:
   Note: You will also need to run this task if you migrated the primary node from a release prior to Version 6.1.0, even if you did not configure a database user registry or a property extension (lookaside) database.
   A. Set the property value for `federated.db.DbType` if using a database user registry or set the property value for `la.DbType` if using a property extension database in the `wkplc.properties` file.
   **Note:** If you migrated the primary node from a version prior to Version 6.1.0 and you are not using a database user
registry or a property extension database, set the property value for `federated.db.DbType` to the same value that is in the `wkplc.properties` file on the primary node.

6. Run the `./ConfigEngine.sh wp-node-prep-vmm-db-secured-environment` task. **Note:** If you want to customize the WebSphere Portal server name (WebSphere_Portal) for your additional node, set the `ServerName` parameter in the `wkplc.properties` file after running the `cluster-node-config-post-federation` task.

7. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell's user registry. Run the `./ConfigEngine.sh wp-change-portal-admin-user` task, from the `wp_profile_root/ConfigEngine` directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

   **Attention:** The password value for `-DWasPassword` is the Deployment Manager administrative password.

   **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

   **Fast path:** If the value for `newAdminGroupId` contains a space; for example Software Group, open the `wkplc.properties` file and add the values for `newAdminId`, `newAdminPw`, and `newAdminGroupId`. Save your changes and run the `ConfigEngine.bat wp-change-portal-admin-user` task.

   **Important:** If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user` task until after the `cluster-node-config-cluster-setup` (static cluster) or `cluster-node-config-dynamic-cluster-setup` (dynamic cluster) task completes. After running the `wp-change-portal-admin-user` task, start or restart the WebSphere_Portal server to use the updated administrator user ID.

   **Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the `wp-change-portal-admin-user` task. `-DnewAdminPw` is an optional parameter to update the Administrative password in the `wkplc.properties` file if required.

8. Optional: Open the `wkplc.properties` file and change the `ServerName` parameter from the default `WebSphere_Portal_nodename` value to a value that meets your business needs. Do not change the parameter to `WebSphere_Portal` as this is the primary node value.

9. Run the `./ConfigEngine.sh cluster-node-config-cluster-setup` task to add the node to your cluster.

10. Wait at least 30 minutes before stopping and restarting the deployment manager because the node agent runs a background thread to expand all EAR files into their target directories. **Note:** Attempting to restart the deployment manager while the expansion is in process can result in an unusable application. Although the time to complete an EAR expansion is environment-dependent, a 30 minute wait is sufficient for a typical environment.

11. Perform the following steps to access the Web Content Management content through an external Web server:
   A. Log on to the deployment manager administrative console.
   B. Select **Environment > WebSphere Variables.**
   C. From the **Scope** drop-down menu, select the `Node=nodename`, `Server=servername` option to narrow the scope of the listed variables, where `Node=nodename` is the node that contains the application server.
D. Update the **WCM_HOST** variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.

E. Update the **WCM_PORT** variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.

F. Perform the following steps to synchronize the node with the deployment manager:
   1. Select **System Administration > Nodes**.
   2. Select the node that you want to synchronize from the list.
   3. Click **Full Resynchronize**.

G. Log off of the deployment manager administrative console.

12. If you entered passwords in any of the properties files while creating your cluster, you should remove them for security purposes. See "Deleting passwords from properties files" under Related tasks for information.

**Parent topic:** Choosing the type of additional node to create on Solaris

**Related tasks**

Deleting passwords from properties files
Adding an additional node to an existing dynamic cluster

After creating your dynamic cluster, you can add additional nodes to expand the capacity of the dynamic cluster. Install and configure the additional node, then add it to the existing dynamic cluster node group, and then add it to the existing IBM® WebSphere® Virtual Enterprise dynamic cluster.

On all WebSphere Portal nodes, install WebSphere Virtual Enterprise and augment the wp_profile profile. See Installing and configuring the product for information about installing WebSphere Virtual Enterprise. See Augmenting profiles for information about augmenting the Deployment Manager and wp_profile profiles.

- Prerequisites
  - PK97393 for all WebSphere Application Server Versions prior to 6.1.0.5
  - Installing and configuring the product:
    - Augmenting profiles

Perform the following steps to add an additional node to an existing WebSphere Virtual Enterprise dynamic cluster:

1. Perform the following steps on the secondary node to access your database server:
   - Note: All properties files are located in the `wp_profile_root/ConfigEngine/properties` directory.
   - For JDBC Type 2 connections only, ensure that the database client software is installed on the secondary node using the same settings as the primary node and that you can connect to the remote database. It is not necessary to create databases or users when configuring secondary nodes. **DB2 users:** When using DB2, you must catalog the TCP/IP node and all databases used on each secondary node. For more information refer to Preparing DB2.
   - For JDBC Type 4 connections only, copy the JDBC Type 4 jar files to the secondary node from the DB2 server.
   - Set all database properties in the `wkplc_dbtype.properties` and `wkplc_comp.properties` files with the appropriate values from the primary node.

2. Run the following tasks to validate your database settings:
   - A. `./ConfigEngine.sh validate-database-driver`
     - `DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password`
   - B. `./ConfigEngine.sh validate-database-connection`
     - `DWasPassword=password`
     - `DTransferDomainList=release,customization,community,jcr,feedback,likeminds`

3. Stop the server1 and WebSphere_Portal servers on the secondary node and ensure that the following parameters are set correctly in the `wkplc.properties` file, located in the `wp_profile_root/ConfigEngine\properties` directory of the secondary node:
   - Note: Although you can add these parameters directly to any task that you run while creating your cluster, you may want to add them to the properties file while creating your cluster and then remove them when you are finished to keep your environment secure.
   - A. Verify that `WasUserId` is set to your deployment manager administrator ID.
   - B. Verify that `WasPassword` is set to your deployment manager administrator password.
   - C. Verify that `PortalAdminId` is set to your WebSphere Portal administrator ID.
   - D. Verify that `PortalAdminPwd` is set to your WebSphere Portal password.
   - E. Verify that `WasRemoteHostName` is set to the fully qualified host name of the deployment manager.
   - F. Verify that `WasSoapPort` is set to the SOAP port that the deployment manager is using; the default value is 8879.
   - G. Verify that `PrimaryNode` is set to `false`. 
H. Verify that ClusterName is set to the primary node's ClusterName.

4. Run the `/ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=dmgr_password` task, from the `wp_profile_root/ConfigEngine` directory of the secondary node.

5. Optional: If you configured a database user registry or a property extension (lookaside) database, run the following task to set up access to the database drivers:
   
   **Note:** You will also need to run this task if you migrated the primary node from a release prior to Version 6.1.0, even if you did not configure a database user registry or a property extension (lookaside) database.
   
   A. Set the property value for `federated.db.DbType` if using a database user registry or set the property value for `la.DbType` if using a property extension database in the `wkplc.properties` file.
   
   **Note:** If you migrated the primary node from a version prior to Version 6.1.0 and you are not using a database user registry or a property extension database, set the property value for `federated.db.DbType` to the same value that is in the `wkplc.properties` file on the primary node.

   B. Run the `/ConfigEngine.sh wp-node-prep-vmm-db-secured-environment -DWasPassword=dmgr_password -DDbDomain=la|federated.db -DVmmNodeName=node_name -Db_type.NodeDbLibrary=local full path of the database jars` task from the `wp_profile_root/ConfigEngine` directory to create the variable used to access the VMM database jars.
   
   **Note:** `VmmNodeName` is a list of one or more WebSphere Portal nodes names in the cell which share the same database driver paths. The `db_type` in `db_type.NodeDbLibrary` should be set to the type of database you are using, for example `db2`.

6. Run the `/ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=dmgr_password` task. **Note:** If you want to customize the WebSphere Portal server name (WebSphere_Portal) for your additional node, set the `ServerName` parameter in the `wkplc.properties` file after running the `cluster-node-config-post-federation` task.

7. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell's user registry. Run the `/ConfigEngine.sh wp-change-portal-admin-user -DWasPassword= -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task, from the `wp_profile_root/ConfigEngine` directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

   **Attention:** The password value for `-DWasPassword` is the Deployment Manager administrative password.

   **Important:** You must provide the full distinguished name (DN) for the `newAdminId` and `newAdminGroupId` parameters.

   **Additional parameter for stopped servers:** This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

   **Fast path:** If the value for `newAdminGroupId` contains a space; for example `Software Group`, open the `wkplc.properties` file and add the values for `newAdminId`, `newAdminPw`, and `newAdminGroupId`. Save your changes and run the `ConfigEngine.bat wp-change-portal-admin-user -DWasPassword=dmgr_password` task.

   **Important:** If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user` task until after the `cluster-node-config-cluster-setup` (static cluster) or `cluster-node-config-dynamic-cluster-setup` (dynamic cluster) task completes. After running the `wp-change-portal-admin-user` task, start or restart the WebSphere_Portal server to use the updated administrator user ID.

   **Note:** The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the `wp-change-portal-admin-user` task. `-DnewAdminPw` is an optional parameter to update the Administrative password in the `wkplc.properties` file if required.

8. Log on to the deployment manager administrative console.

9. Perform the following steps to add members to the node group:
   
   A. Click **System administration > Node groups.**
B. Click on the name of the node group that you want to add members to.
C. Click Node group members under Additional Properties.
D. Click Add.
E. Select the additional node you want to add into the node group and then click Add.
F. Click the Save link to save your changes to the master configuration.

10. Wait at least 30 minutes before stopping and restarting the deployment manager because the node agent runs a background thread to expand all EAR files into their target directories. **Note:** Attempting to restart the deployment manager while the expansion is in process can result in an unusable application. Although the time to complete an EAR expansion is environment-dependent, a 30 minute wait is sufficient for a typical environment.

11. Define or verify the following parameters in the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory:

   A. Set ClusterName to the name of the existing dynamic cluster.
   B. Verify that CellName is set to the deployment manager cell.
   C. Verify that NodeName is set to the local WebSphere Portal node.
   D. Set ServerName to the server that will be used for the dynamic cluster member on this node. **Note:** Log on to the deployment manager administrative console and click Dynamic Clusters > PortalCluster > Dynamic cluster members to find the name of the server used for the dynamic cluster.
   E. Verify that PrimaryNode is set to false because this is an additional node.

12. Run the `./ConfigEngine.sh cluster-node-config-dynamic-cluster-setup -DWasPassword=dmgr_password` task to add the new member to the existing dynamic cluster. **Note:** This task may display a warning message since the cluster already exists but it will proceed to create the new dynamic cluster; therefore, the warning can be ignored.

13. Perform the following steps to start the cluster member:

   A. Log on to the Deployment Manager administrative console.
   B. Navigate to Servers > Dynamic clusters > cluster name > Dynamic cluster members.
   C. Select the cluster member and click Start.
   D. Log off of the deployment manager administrative console.

14. Perform the following steps to access the Web Content Management content through an external Web server:

   A. Log on to the deployment manager administrative console.
   B. Select Environment > WebSphere Variables.
   C. From the Scope drop-down menu, select the Node=nodename, Server=servername option to narrow the scope of the listed variables, where Node=nodename is the node that contains the application server.
   D. Update the WCM_HOST variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.
   E. Update the WCM_PORT variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.
   F. Perform the following steps to synchronize the node with the deployment manager:
      1. Select System Administration > Nodes.
      2. Select the node that you want to synchronize from the list.
      3. Click Full Resynchronize.
   G. Log off of the deployment manager administrative console.

15. Run the following tasks to propagate your changes. **Note:** WebSphere_Portal_nodename is the name of the node’s WebSphere Portal server; but if you customized the server name, the default name changes. If you are uncertain of the server name, run the `serverstatus -all` task to get a list of the server names and their status.

   A. `./stopManager.sh -username admin_userid -password admin_password`, from the dmgr_profile_root/bin directory
   B. `./stopNode.sh -username admin_userid -password admin_password`, from the wp_profile_root/bin directory
   C. `./stopServer.sh WebSphere_Portal_nodename -username admin_userid -password admin_password`
D. ./startManager.sh, from the dmgr_profile_root/bin directory
E. ./startNode.sh, from the wp_profile_root/bin directory
F. ./startServer.sh WebSphere_Portal_nodename

16. If you entered passwords in any of the properties files while creating your cluster, you should remove them for security purposes. See "Deleting passwords from properties files" under Related tasks for information.

**Parent topic:** Choosing the type of additional node to create on Solaris

**Related tasks**
Deleting passwords from properties files
Choosing the type of vertical cluster to create on Solaris

If you are using the IBM® WebSphere® Application Server Network Deployment, you can add vertical cluster members to your static cluster. If you are using IBM WebSphere Virtual Enterprise, you can add vertical cluster members to your dynamic cluster.

Choose one of the following options to add vertical cluster members to your cluster.

- Adding vertical cluster members to a static cluster on Solaris
  You can add vertical cluster members to share the workload demands of your cluster across multiple members running on the same physical machine.

- Adding vertical cluster members to a dynamic cluster on Solaris
  You can add vertical cluster members to share the workload demands of your cluster across multiple members running on the same physical machine.

*Parent topic:* Preparing additional nodes on Solaris
Adding vertical cluster members to a static cluster on Solaris

You can add vertical cluster members to share the workload demands of your cluster across multiple members running on the same physical machine.

Perform the following steps to add a vertical cluster member to your clustered environment:

1. Log into the deployment manager administrative console.
2. Depending on your version of WebSphere Application Server, make the appropriate selection from the navigation:
   - For **WebSphere Application Server Version 6.1**: Click **Servers > Clusters** in the console navigation tree, select the cluster name, and then click **Cluster members** from the list of additional properties.
   - For **WebSphere Application Server Version 7.0**: Click **Servers > Clusters > WebSphere application server clusters > cluster_name > Cluster members**.
3. Click **New** to create the cluster member.
   - A. Define the name of cluster member. **Note:** Do not use spaces in the cluster member name.
   - B. Select an existing node where IBM® WebSphere® Portal is installed.
   - C. Check the box **Generate Unique HTTP Ports**.
   - D. Click **Add Member** and then click **Next** to view the summary.
     **Important:** You must update the virtual host entries for the new port created when adding a cluster member. You can do this by updating the default_host virtual host in the administrative console and adding a new alias entry for the port number (use an "*" wildcard character for the host name). See Configuring virtual hosts for information.
4. Click **Finish**, and save the changes.
   - The new cluster topology can be viewed from the **Servers > Cluster Topology** view. If using WebSphere Application Server Version 7.0, the new cluster topology can be viewed from the **Servers > Clusters > Cluster Topology** view.
   - The **Servers > Application Servers** view will list the new server cluster members. If using WebSphere Application Server Version 7.0, the **Servers > Server Types > WebSphere application servers** view will list the new server cluster members.
5. Perform the following steps to enable cache replication:
   - A. From the deployment manager Administrative Console, navigate to **Servers > Application servers** and then click the new vertical cluster member(s). If using WebSphere Application Server Version 7.0, navigate to **Servers > Server Types > WebSphere application servers** and then click the new vertical cluster member(s).
   - B. Click **Dynamic cache service** under **Container services**.
   - C. Change **Cache size** to 3000 entries.
   - D. Check the **Enable cache replication** check box.
   - E. Select **NOT_SHARED** from the **Replication type** drop-down menu.
   - F. Click **OK**.
   - G. Click **Save** to save your changes to the master configuration.
6. Perform the following steps on each vertical cluster member to clean up the server-scoped resources, caches, and resource providers:
   - A. Run the `./ConfigEngine.sh cluster-node-config-vertical-cluster-setup -DServerName=unique vertical cluster servername -DWasPassword=password` task, from the **wp_profile_root/ConfigEngine** directory, where **unique vertical cluster servername** is the name you specified when you created the cluster member.
   - B. Restart the vertical cluster member referenced in the `cluster-node-config-vertical-cluster-setup` task.
7. Perform the following steps to access the Web Content Management content through an external Web server:
   A. Log on to the deployment manager administrative console.
   B. Select Environment > WebSphere Variables.
   C. From the Scope drop-down menu, select the Node=nodename, Server=servername option to narrow the scope of the listed variables, where Node=nodename is the node that contains the application server.
   D. Update the WCM_HOST variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.
   E. Update the WCM_PORT variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.
   F. Perform the following steps to synchronize the node with the deployment manager:
      1. Select System Administration > Nodes.
      2. Select the node that you want to synchronize from the list.
      3. Click Full Resynchronize.
   G. Log off of the deployment manager administrative console.

8. Save your changes and resynchronize the nodes.
   A. In the administrative console for the deployment manager, click Save on the task bar, and save your administrative configuration.
   B. Select System Administration > Nodes, select the node from the list, and click Full Resynchronize.

9. Regenerate the Web server plug-in.
   A. Regenerate the Web server plug-in using the deployment manager administrative console.
   B. If you are using a remote Web server, copy the updated plug-in configuration file (plugin-cfg.xml) to the Web server’s plug-in configuration directory.

10. Stop and start the Web server.

Parent topic: Choosing the type of vertical cluster to create on Solaris
Adding vertical cluster members to a dynamic cluster on Solaris

You can add vertical cluster members to share the workload demands of your cluster across multiple members running on the same physical machine.

Perform the following steps to add a vertical cluster member to your clustered environment:

1. Open a browser and enter http://DM01:9060/ibm/console in the address bar to access the administrative console on the deployment manager, where DM01 is the deployment manager node or host name. The port number might differ based on your installation.

2. Perform the following steps to allow vertical clusters on your dynamic cluster:
   A. Navigate to Servers > Dynamic clusters and select the appropriate dynamic cluster.
   B. Select the Allow more than one instance to start on the same node check box under Vertical stacking of Instances on node.
   C. Enter a new value in the Number of instances text box to determine the number of vertical cluster members allowed on each node.
   D. Click Apply and then click Save to save the changes to the master configuration.

   - The new cluster topology can be viewed from the Servers > Cluster Topology view. If using WebSphere Application Server Version 7.0, the new cluster topology can be viewed from the Servers > Clusters > Cluster Topology view.
   - The Servers > Application Servers view will list the new server cluster members. If using WebSphere Application Server Version 7.0, the Servers > Server Types > WebSphere application servers view will list the new server cluster members.

   **Important:** You must update the virtual host entries for the new port created when adding a cluster member. You can do this by updating the default_host virtual host in the administrative console and adding a new alias entry for the port number (use an “*” wildcard character for the host name). See Configuring virtual hosts for information.

3. Perform the following steps to enable cache replication:
   A. From the deployment manager Administrative Console, navigate to Servers > Application servers and then click the new vertical cluster member(s). If using WebSphere Application Server Version 7.0, navigate to Servers > Server Types > WebSphere application servers and then click the new vertical cluster member(s).
   B. Click Dynamic cache service under Container services.
   C. Change Cache size to 3000 entries.
   D. Check the Enable cache replication check box.
   E. Select NOT_SHARED from the Replication type drop-down menu.
   F. Click OK.
   G. Click Save to save your changes to the master configuration.

4. Perform the following steps on each vertical cluster member to clean up the server-scoped resources, caches, and resource providers:
   A. Run the ./ConfigEngine.sh cluster-node-config-vertical-cluster-setup -DServerName=unique vertical cluster servername -DWasPassword=password task, from the wp_profile_root/ConfigEngine directory, where unique vertical cluster servername is the name you specified when you created the cluster member.
   B. Restart the vertical cluster member referenced in the cluster-node-config-vertical-cluster-setup task.

5. Perform the following steps to access the Web Content Management content through an external Web server:
   A. Log on to the deployment manager administrative console.
B. Select **Environment > WebSphere Variables**.

C. From the **Scope** drop-down menu, select the `Node-nodename, Server-servername` option to narrow the scope of the listed variables, where `Node-nodename` is the node that contains the application server.

D. Update the **WCM_HOST** variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router.

E. Update the **WCM_PORT** variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router.

F. Perform the following steps to synchronize the node with the deployment manager:
   1. Select **System Administration > Nodes**.
   2. Select the node that you want to synchronize from the list.
   3. Click **Full Resynchronize**.

G. Log off of the deployment manager administrative console.

6. Save your changes and resynchronize the nodes.
   
   A. In the administrative console for the deployment manager, click **Save** on the task bar, and save your administrative configuration.
   
   B. Select **System Administration > Nodes**, select the node from the list, and click **Full Resynchronize**.

7. Stop and start the Web server.

**Parent topic:** Choosing the type of vertical cluster to create on Solaris
Tune your servers

Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on Solaris
  - Preparing your Solaris operating system
  - Preparing the primary node on Solaris
  - **Base Portal Tuning Scenarios**

Tuning a WebSphere Portal environment involves tuning and configuring the various systems and components of the environment. The tuning guide provides general concepts and details specifics configuration instructions. Instructions are included for:

- Configuring the application server and the resources defined for that application server
- Determining the cloning strategy for expanding or extending the environment
- Tuning the database(s) and database server
- Tuning the directory server and its database
- Tuning the Web server
- Tuning the operating system and network
- Tuning the WebSphere Portal services

**Parent topic:** Setting up a cluster on Solaris

**Previous topic:** Preparing additional nodes on Solaris

**Next topic:** Configuring search in a cluster on Solaris

**Related information**
- WebSphere Portal and Lotus Web Content Management 6.1.x Performance Tuning Guide
- Tuning a Cluster Environment
- IBM WebSphere Portal Performance Troubleshooting Guide
Configuring search in a cluster on Solaris

IBM® WebSphere® Portal provides two distinct search capabilities. You can utilize both types of search capabilities in a clustered environment.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on Solaris
  - Preparing your Solaris operating system
  - Preparing the primary node on Solaris
  - Tune your servers

- **Configuring Portal Search in a cluster on Solaris**
  To support Portal Search in a clustered environment, you must install and configure search for remote search service on an IBM WebSphere Application Server node that is not part of the IBM WebSphere Portal cluster.

- **Configuring JCR search in a cluster on Solaris**
  To enable search in a cluster for content stored in the JCR database, you must configure each server in the cluster to access a shared directory. JCR-based content includes content created with Web Content Management or Personalization.

**Parent topic:** Setting up a cluster on Solaris  
**Previous topic:** Tune your servers  
**Next topic:** Setting up multiple clusters on Solaris
Configuring Portal Search in a cluster on Solaris

To support Portal Search in a clustered environment, you must install and configure search for remote search service on an IBM® WebSphere® Application Server node that is not part of the IBM WebSphere Portal cluster.

To install and configure the search service remotely, perform the following tasks:

1. Install and configure the search service to work remotely, that is, on a remote WebSphere Application Server node which is not part of the portal cluster. You can provide the remote search service either as an EJB or as a Web service via SOAP. Deploy the appropriate EJB or SOAP EAR file on the remote WebSphere Application Server node. For details about how to do this, refer to the WebSphere Application Server documentation.

2. Configure the search portlets for remote search service so that they access the remote server accordingly.

Notes:

1. If you have configured a remote search service for a portal cluster, you need to configure the default location for search collections to a directory on the remote server that has write access.

2. The portal site default search collection is created only once at the first time when an administrator selects the search administration portlet Manage Search. If this occurred before you configure the portlet for remote search, then the default portal site search collection is only available on the primary node of the cluster, but not on the remote server. In this case you need to recreate the portal site collection to make it available for search on all nodes of the cluster.
Configuring JCR search in a cluster on Solaris

To enable search in a cluster for content stored in the JCR database, you must configure each server in the cluster to access a shared directory. JCR-based content includes content created with Web Content Management or Personalization.

Create a shared directory called `jcr/search` on a server in the network and ensure that each node in the cluster and the Deployment Manager has network access to the directory.

**Note:** If you are creating content in a clustered environment using the authoring portlet provided with Web Content Management, additional configuration steps are required to enable content created by these content features to be searchable in a cluster.

Perform the following steps on each server in the cluster to configure Search in a clustered environment:

1. **Edit the** `icm.properties` **file, located in the** `wp_profile_root/PortalServer/jcr/lib/com/ibm/icm` **directory.**
2. **Change the value of the** `jcr.textsearch.indexdirectory` **property to the shared directory; for example,**
   
   ```bash
   jcr.textsearch.indexdirectory=\\your_server\your_share\jcr\search
   ```

   You can specify the shared directory value in one of the following formats: *Table 1. The format for the shared directory.*

<table>
<thead>
<tr>
<th>Format</th>
<th>Shared directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal Naming Convention (UNC) format</td>
<td><code>\\your_server\your_share\jcr\search</code></td>
</tr>
<tr>
<td>Example:</td>
<td><code>\\hostname.example.com\share\jcr\search</code></td>
</tr>
<tr>
<td>Mounted resource format (with forward slashes)</td>
<td><code>/your_share/jcr/search</code></td>
</tr>
<tr>
<td>Example:</td>
<td><code>/mnt/jcr/search</code></td>
</tr>
</tbody>
</table>

   **Important:** This format requires that you mount the shared directory to the local server (for example, through a mapped network drive or a mounted directory). When using the mounted resource format, always use forward slashes instead of back slashes, regardless of the native operating system path format.

3. **Required:** Perform the following steps to delete the default search collections from the Manage Search portlet:
   
   A. Log on to WebSphere Portal as an administrator.
   B. Click **Administration > Search Administration > Manage Search.**
   C. Click **Search Collections.**
   D. Click the **Delete Collection** icon for the **Portal Content** search collection.
   E. Click **OK.**
   F. Restart the WebSphere_Portal server.
   G. Go to the Manage Search portlet and confirm that the **Portal Content** search collection was deleted.

**Parent topic:** Configuring search in a cluster on Solaris
Setting up multiple clusters on Solaris

The majority of the steps to build another cluster are the same as when building the first cluster in the same cell, with a few exceptions. Basically, the new profile will be designated as the primary profile, using IBM® WebSphere® Portal clustering terminology, and will then be used as the basis for the new cluster definition. This duplicates the build process of the first cluster in the cell. During the federation process, if any applications on this new node already exist in the cell (because they are in use by the first cluster), then Deployment Manager will not allow them to be added. After federation, the applications that already exist in the cell are not mapped to the WebSphere_Portal server on the newly federated node, and thus the existing applications must be remapped to this newly federated server to restore its application list. Therefore, depending on the configuration of the new profile, there will likely be some combination of applications shared with the other existing cluster, and some applications unique to this new profile.

Before attempting alternative approaches for building multiple portal-based clusters within a single cell, please contact IBM.

- Prerequisites
  - Preparing prerequisite and corequisite software on Solaris
  - Preparing your Solaris operating system
  - Preparing the primary node on Solaris
  - Tune your servers

- Installing multiple clusters in a single cell on Solaris
  Create a new, independent IBM WebSphere Portal cluster in a cell where a WebSphere Portal cluster already exists.

- Routing requests across clusters on Solaris
  The HTTP Server plug-in that comes with IBM WebSphere Application Server is typically used to balance requests for applications across members of the cluster. While an application can be mapped to more than one cluster, automatic plug-in generation does not provide routing or balancing traffic for the same application across multiple clusters. Multiple cluster environments with shared applications therefore cannot rely solely on WebSphere Application Server automatic plug-in generation to be able to route requests using a web server. One option in this scenario is developing a customized method for defining and maintaining the plug-in configuration file used by the Web server to provide for the required routing of user requests.

Parent topic: Setting up a cluster on Solaris
Previous topic: Configuring search in a cluster on Solaris
Next topic: Sharing database domains between clusters on Solaris
Installing multiple clusters in a single cell on Solaris

Create a new, independent IBM® WebSphere® Portal cluster in a cell where a WebSphere Portal cluster already exists. In the following steps, Cluster A will be used to describe the existing cluster. Portal B will be used to describe the new server profile that will be the basis for the new cluster definition, Cluster B. Perform the following steps to install multiple clusters in a single cell:

1. Upgrade Cluster A, including the Deployment Manager node, to the current, supported hardware and software levels and to the current version of IBM WebSphere Portal.
2. Install and configure Portal B; see the "Preparing the primary node" topic for the appropriate operating system for details. Important: Maintain the same number of data sources with identical names to the Cluster A data sources so that data source bindings in the applications can be resolved on every cluster in which they run. If implementing database sharing across the clusters, the above statement refers to both the shared and non-shared domains; all domains should use the same names.
3. Optional: Using the same database user ID and password for each identically named domain/data source will allow the existing JAAS Authentication Aliases to be functional. If unique database user ID and password are required, additional manual configuration is needed to create new JAAS Authentication Aliases for each data source and map these accordingly. On the primary node of Cluster A, run the ./ConfigEngine.sh create-alias-multiple-cluster -DauthDomainList=release,jcr -DWasPassword=dmgr_password task from the wp_profile_root/ConfigEngine directory to create the new JAAS Authentication Aliases. where authDomainList is set to a list of domains which use unique database user ID and passwords and those domain properties are set correctly in the wkplc_comp.properties file, including user ID and password.
4. Optional: If necessary, upgrade Portal B to the current cumulative fix.
5. Run the ./ConfigEngine.sh mapped-app-list-create -DWasPassword=password task from the wp_profile_root/ConfigEngine directory to build an inventory list of Portal B enterprise applications and portlets.
6. Stop the server1 and WebSphere_Portal servers on Portal B and ensure that the following parameters are set correctly in the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory:
   - Set WasSoapPort to the deployment manager's port.
   - Set WasRemoteHostName to the full host name of the deployment manager.
7. If the Deployment Manager is configured to use a stand-alone LDAP user registry, update the wkplc.properties file, located in the wp_profile_root/ConfigEngine/properties directory, on the primary node with the stand-alone LDAP user registry property values from the Deployment Manager. You can find these settings under the VMM Stand-alone LDAP configuration heading. Important: Ensure that you set WasUserId and WasPassword to the Deployment Manager user ID and password.
8. Run the ./ConfigEngine.sh cluster-node-config-pre-federation -DWasPassword=password task, from the wp_profile_root/ConfigEngine of the primary node.
9. Run the ./ConfigEngine.sh map-apps-to-server -DWasPassword=password task to determine which applications from the inventory list are no longer mapped to Portal B. The task uses the application profiles already in the cell to restore the mappings. Wait 30 minutes after running this task to allow all EAR files to expand before proceeding to the next step.
10. Perform the following steps to federate Portal B into the deployment manager cell:
    A. Ensure that all database parameters are correctly set, including passwords, in the wkplc_comp.properties and wkplc_dbtype.properties files.
B. Run the `./ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=dmgr_password` task.

C. After running the `cluster-node-config-post-federation` task, wait at least 30 minutes to allow all EAR files to expand.

D. Since the WebSphere Portal node is now using security settings from the Deployment Manager cell, you need to update the WebSphere Portal administrative user ID and password to match an administrative user defined in the cell’s user registry. Run the `./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=password -DnewAdminId=newadminid -DnewAdminPw=newpassword -DnewAdminGroupId=newadmingroupid` task, from the `wp_profile_root/ConfigEngine` directory, to update the WebSphere Portal administrative user ID if the Deployment Manager cell is using a different user registry.

Additional parameter for stopped servers: This task verifies the user against a running server instance. If the server is stopped, add the `-Dskip.ldap.validation=true` parameter to the task to skip the validation.

Important: If standalone LDAP security is already enabled on the Deployment Manager cell, delay running the `wp-change-portal-admin-user` task until after the `cluster-node-config-cluster-setup` (static cluster) or `cluster-node-config-dynamic-cluster-setup` (dynamic cluster) task completes. After running the `wp-change-portal-admin-user` task, start or restart the WebSphere Portal server to use the updated administrator user ID.

Note: The WebSphere Portal administrative user ID and administrative group must exist in the Deployment Manager before running the `wp-change-portal-admin-user` task. `-DnewAdminPw` is an optional parameter to update the Administrative password in the wkplc.properties file if required.

E. From the administrative console, click System Administration > Node Agents.

F. Check the box next to the required node agent and then click Restart.

G. Stop and restart the deployment manager.

H. Stop and restart the WebSphere Portal server on Portal B

11. Restart the WebSphere Portal server on Cluster A. Verify that Cluster A is functionally intact by spot checking pages and portlets and then verify that Portal B is functionally intact by spot checking pages and portlets that you deployed into Portal B before it was federated. Any discrepancies or errors should be corrected now before continuing. Note: If Portal B is using a non-default Portal server administrative ID, not wpsadmin, the server will not be functional until the cluster configuration is complete and the Portal administrative ID has been configured to match the Cells security settings.

12. Choose one of the following options to define a cluster using Portal B as the basis:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
Perform the following steps to define a static cluster using Portal B as the basis: Run the 
`./ConfigEngine.sh cluster-node-config-cluster-setup -DWasPassword=dmgr_password`
task. Configure the cluster to use an external Web server to take advantage of features such as workload management. Choose one of the following options: Configuring a Web server and an application server on separate machines (remote) Configuring a Web server and an application server profile on the same machine Configuring a Web server and a deployment manager profile on the same machine

**Note:** Start with the step about launching the Plug-ins installation wizard. Perform the following steps to access the Web Content Management content through an external Web server: Log on to the deployment manager administrative console. Select Environment > WebSphere Variables. From the Scope drop-down menu, select the Node=nodename, Server=servername option to narrow the scope of the listed variables, where Node=nodename is the node that contains the application server. Update the **WCM_HOST** variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router. Update the **WCM_PORT** variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router. Save your changes and then restart the deployment manager, the node agent(s), server1, and the WebSphere_Portal servers. Change the task list settings to point to the cluster instead of a single server if using Portal business process integration. See one of the following sections in the Information Center for information: Cross cell: Adding a BPI-enabled portal server to a managed cell in a cross-cell setup Single cell: Adding a BPI-enabled portal server to a managed cell in a single-cell setup
Perform the following steps to define a dynamic cluster using Portal B as the basis:

Log on to the deployment manager administrative console.

Perform the following steps to create a node group:

Click **New**. Type the node group **Name**. Type any information about the node group in the **Description** text box. Click **OK**. Click the **Save** link to save your changes to the master configuration.

Perform the following steps to add members to the node group:

Click **System administration > Node groups**. Click on the name of the node group that you want to add members to. Click **Node group members** under **Additional Properties**. Click **Add**. Select the primary node and then click **Add**. Click the **Save** link to save your changes to the master configuration.

Perform the following steps to create a dynamic cluster in the node group:

Click **Servers > Dynamic clusters**. Click **New**. Select WebSphere Application Server from the **Server Type** pull-down menu and then click **Next**. Type the cluster name in the **Dynamic cluster name** text box and then click **Next**. Type the same value that you provided for the **ClusterName** parameter in the **wkplc.properties** file of your primary node.

Remove all default membership policies and then click **Subexpression builder**. Enter the following information in the Subexpression builder window:

- Select **Logical operator** pull-down menu.
- Select **Nodegroup** from the **Select operand** pull-down menu.
- Select **Equals (=)** from the **Operator** pull-down menu.
- Type the nodegroup name you created in the previous step in the **Value** text box. Click **Generate subexpression**. Click **Append**.
- Click **Preview membership** to verify that all nodes included in the nodegroup display and then click **Next**. Click the **Create the cluster member using an existing server as a template** radio button and then select the WebSphere_Portal server for the primary node from the pull-down menu. Click **Next**. Specify the dynamic cluster properties and then click **Next**.
- Review the summary page to verify your actions and then click **Finish**.

Define or verify the following parameters in the **wkplc.properties** file, located in the **wp_profile_root/ConfigEngine/properties** directory:

- Set **ClusterName** to the name of the new dynamic cluster.
- Verify that **CellName** is set to the deployment manager cell.
- Verify that **NodeName** is set to the local WebSphere Portal node.
- Set **ServerName** to the server that will be used for the dynamic cluster member on this node.
- Verify that **PrimaryNode** is set to true.

Run the **. /ConfigEngine.sh cluster-node-config-dynamic-cluster-setup - DWasPassword=dmgr_password** task from the **wp_profile_root/ConfigEngine** directory to create the dynamic
13. Install any additional nodes to the cell to support additional cluster members for **Cluster B** identically to the primary node, and then federate as them as secondary nodes and define as cluster members on these nodes. For information about adding additional nodes navigate to **Installing WebSphere Portal > Setting up WebSphere Portal > Setting up a clustered production environment**. Select **Environment > WebSphere Variables**. From the **Scope** drop-down menu, select the **Node=nodename, Server=servername** option to narrow the scope of the listed variables, where **Node=nodename** is the node that contains the application server. Update the **WCM_HOST** variable with the fully qualified host name used to access the WebSphere Portal server through the Web server or On Demand Router. Update the **WCM_PORT** variable with the port number used to access the WebSphere Portal server through the Web server or On Demand Router. Save your changes and then restart the deployment manager, the node agent(s), server1, and the WebSphere Portal servers. If you are using a Web server to connect to the On Demand Router (ODR), configure the web server as a trusted proxy on the ODR. Refer to **Configuring a Web server as a trusted proxy server** for instructions. **Tip:** You can also configure the ODR to dynamically update the Web server configuration when changes occur. Refer to **Configuring an on demand router to dynamically update the Web server plug-in configuration** for instructions. Change the task list settings to point to the cluster instead of a single server if using Portal business process integration. See one of the following sections in the Information Center for information: **Cross cell: Adding a BPI-enabled portal server to a managed cell in a cross-cell setup** **Single cell: Adding a BPI-enabled portal server to a managed cell in a single-cell setup**

14. Restart the server1 and WebSphere_Portal servers on **Cluster A** and **Cluster B**.

15. If you entered passwords in any of the properties files while creating your cluster, you should remove them for security purposes. See “Deleting passwords from properties files” under Related tasks for information.

Installation of **Cluster B** is complete. It is now an independent cluster from **Cluster A**, which means that **Cluster B** can have its own configuration, set of end-user portlets, and target community. Any applications that are common between **Cluster A** and **Cluster B** are most likely infrastructure or related to administration, and special care needs to be taken to preserve their commonality between clusters and correct maintenance levels.

**Parent topic:** Setting up multiple clusters on Solaris

**Related tasks**
Deleting passwords from properties files
Adding a BPI-enabled portal server to a managed cell in a cross-cell setup
Adding a BPI-enabled portal server to a managed cell in a single-cell setup
Routing requests across clusters on Solaris

The HTTP Server plug-in that comes with IBM® WebSphere® Application Server is typically used to balance requests for applications across members of the cluster. While an application can be mapped to more than one cluster, automatic plug-in generation does not provide routing or balancing traffic for the same application across multiple clusters. Multiple cluster environments with shared applications therefore cannot rely solely on WebSphere Application Server automatic plug-in generation to be able to route requests using a web server. One option in this scenario is developing a customized method for defining and maintaining the plug-in configuration file used by the Web server to provide for the required routing of user requests.

An important consideration in a multiple cluster environment is ensuring that all subsequent HTTP requests for an end user are routed to the same cluster that processed the first HTTP request. The WebSphere Portal login processing depends upon preserving this cluster affinity during this initial time until the user has successfully logged in and session cookies maintain affinity. In order to guarantee that affinity is preserved during login, set the Navigator Service `public.session` parameter to a value of true. Refer to “Portal Configuration Services” for information on how to configure this parameter.

**Parent topic:** Setting up multiple clusters on Solaris
Sharing database domains between clusters on Solaris

To help provide redundancy and failover support in production environments composed of multiple clusters in a single cell and multiple clusters in different cells, you can share database domains between those clusters. IBM® WebSphere® Portal data is organized into different database domains, with different availability requirements depending on how the production environment is set up. When multiple lines of production are involved and each line of production is implemented as a cluster of servers, sharing database domains ensures that the data is automatically synchronized across the lines of production.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on Solaris
  - Preparing your Solaris operating system
  - Preparing the primary node on Solaris
  - Tune your servers

**Important:** JCR domains and release domains cannot be shared between different Web Content Management clusters or servers. Each distinct cluster or server in your Web Content Management system must use a separate JCR or release domain. For example:

<table>
<thead>
<tr>
<th>Development Server</th>
<th>Authoring Cluster</th>
<th>Staging Server</th>
<th>Delivery Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCR Domain 1</td>
<td>JCR Domain 2</td>
<td>JCR Domain 3</td>
<td>JCR Domain 4</td>
</tr>
<tr>
<td>Release Domain 1</td>
<td>Release Domain 2</td>
<td>Release Domain 3</td>
<td>Release Domain 4</td>
</tr>
</tbody>
</table>

Perform the following steps to share database domains when setting up an environment with multiple clusters:

1. Set up the first cluster (referred to as Cluster A in these instructions).
2. Determine which database domains you want to share with any other clusters in the environment.
3. Install the primary node of the next cluster (Cluster B), and perform the following steps to configure the node to use the shared database domains.
   A. Perform a partial database transfer of the database domains that you are not sharing. For example, if you are sharing only the Customization and Community domains, you would transfer the remaining domains to the database you are using for Cluster B.
   B. Reconfigure the shared database domains on the node to connect to the shared database domains you are using for Cluster A. For example, to connect to the Customization and Community domains for Cluster A, you would connect to those domains from Cluster B.
4. Continue setting up the primary node as described in the cluster instructions.
5. Install the secondary node of Cluster B, and perform the following steps to configure the node to use the shared database domains.
   A. For those database domains that you are not sharing between clusters, reconfigure the domains to connect to the database domains you are using for Cluster B. As in the example for the primary node, if you are sharing only the Customization and Community domains, reconfigure the remaining domains on the secondary node to use the domains of the primary node for Cluster B.
   B. Reconfigure the shared database domains on the node to connect to the shared database domains you are using for Cluster A. For example, to connect to the Customization and Community domains for Cluster A, you would connect to
6. Continue setting up the secondary node as described in the cluster instructions.

**Parent topic:** Setting up a cluster on Solaris  
**Previous topic:** Setting up multiple clusters on Solaris

**Related tasks**  
Connecting to existing database domains
Setting up a cluster on Windows

In a production environment, you can install and configure IBM® WebSphere® Portal on a single (primary) node and then add additional nodes in the future as your company's needs grow. You can also deploy a high availability production environment by creating and configuring clustered deployments of multiple servers running WebSphere Portal managed by IBM WebSphere Application Server Network Deployment.

- Prerequisites
  - [Technotes for installation and configuration issues](#)

Perform the following steps to set up your production environment on Windows:

1. **Preparing prerequisite and corequisite software on Windows**
   Before creating your high availability production environment, you must prepare your database, your user registry, your deployment manager, and your remote Web server.

2. **Preparing your Windows operating system**
   View information on setting up your operating system for IBM WebSphere Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

3. **Preparing the primary node on Windows**
   Before creating your high availability environment, you must install IBM WebSphere Portal on your primary node and then configure your database and your network deployment manager.

4. **Choosing the type of cluster to create on Windows**
   If you installed a IBM WebSphere Application Server Network Deployment, create a static cluster to handle failover requests. If you installed a IBM WebSphere Virtual Enterprise, create a dynamic cluster to balance member workloads.

5. **Preparing a remote Web server when portal is installed on Windows**
   Install and configure the Web server plug-in, provided by IBM WebSphere Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

6. **Configuring WebSphere Portal to use a user registry on Windows in a clustered environment**
   Configure user registry security on IBM WebSphere Portal to protect your server from unauthorized users. You can configure a stand-alone LDAP user registry or you can add LDAP user registries and/or database user registries to the default federated repository. After configuring your user registry, you can add realms for Virtual Portals or a lookaside database to store attributes that cannot be stored in the LDAP user registry.

7. **Preparing additional nodes on Windows**
   After installing and configuring your primary node, you can create your secondary nodes. You must install IBM WebSphere Portal on each node and then configure the node to access the database and user registry before adding it to the cluster.

8. **Tune your servers**
   Tuning the servers is important to the performance of your WebSphere Portal environment. WebSphere Portal is not tuned for a production environment out of the box, so to ensure optimal performance, review and complete the steps in the IBM WebSphere Portal Tuning Guide. Even if a tuning guide is not available for the current release of WebSphere Portal, the tuning guide for the previous product version should be used.

9. **Configuring search in a cluster on Windows**
   IBM WebSphere Portal provides two distinct search capabilities. You can utilize both types of search capabilities in a clustered environment.
10. **Setting up multiple clusters on Windows**

   The majority of the steps to build another cluster are the same as when building the first cluster in the same cell, with a few exceptions. Basically, the new profile will be designated as the primary profile, using IBM WebSphere Portal clustering terminology, and will then be used as the basis for the new cluster definition. This duplicates the build process of the first cluster in the cell. During the federation process, if any applications on this new node already exist in the cell (because they are in use by the first cluster), then Deployment Manager will not allow them to be added. After federation, the applications that already exist in the cell are not mapped to the WebSphere_Portal server on the newly federated node, and thus the existing applications must be remapped to this newly federated server to restore its application list. Therefore, depending on the configuration of the new profile, there will likely be some combination of applications shared with the other existing cluster, and some applications unique to this new profile.

11. **Sharing database domains between clusters on Windows**

   To help provide redundancy and failover support in production environments composed of multiple clusters in a single cell and multiple clusters in different cells, you can share database domains between those clusters. IBM WebSphere Portal data is organized into different database domains, with different availability requirements depending on how the production environment is set up. When multiple lines of production are involved and each line of production is implemented as a cluster of servers, sharing database domains ensures that the data is automatically synchronized across the lines of production.

**Parent topic:** Setting up a cluster
Preparing prerequisite and corequisite software on Windows

Before creating your high availability production environment, you must prepare your database, your user registry, your deployment manager, and your remote Web server.

Perform the following tasks to prepare prerequisite and corequisite software:

1. Preparing the WebSphere Application Server Deployment Manager on Windows
   IBM WebSphere Portal provides a customized installation package (CIP) that includes IBM WebSphere Application Server Network Deployment and all required maintenance packages. Use the supplied discs to install WebSphere Application Server Network Deployment on a dedicated system.

2. Preparing a remote Web server when portal is installed on Windows
   Install and configure the Web server plug-in, provided by IBM WebSphere Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

Parent topic: Setting up a cluster on Windows
Next topic: Preparing your Windows operating system
Preparing the WebSphere Application Server Deployment Manager on Windows

IBM® WebSphere® Portal provides a customized installation package (CIP) that includes IBM WebSphere Application Server Network Deployment and all required maintenance packages. Use the supplied discs to install WebSphere Application Server Network Deployment on a dedicated system.

Perform the following steps to prepare the WebSphere Application Server Deployment Manager:

1. Install WebSphere Application Server Network Deployment or use the CIP to upgrade an existing installation. Run the following command:
   ```markdown
cd_root\windows\ia32\ifpackage\WAS\install.exe
or
cd_root\windows\amd64\ifpackage\WAS\install.exe
```
   where `cd_root` is the root directory of the disc.

2. Choose one of the following options to create a default deployment manager profile:
   **Important:** While creating the default deployment manager profile, enable administrative security. If you use the Profile Management Tool, check the enable administrative security check box. If you use the manageprofile command, add the `-enableAdminSecurity=true` parameter to the command line.

   - **Profile Management Tool**
     See [Creating a deployment manager profile](#) for information.

   - **manageprofile commands**
     See [manageprofile commands](#) for information.
     **Note:** If you have a 64-bit environment, only the manageprofiles command is supported when creating profiles.

3. Run the following command to start the deployment manager: `startManager.bat`, from the `dmgr_profile_root\bin` directory.

4. Use the following URL to launch the network deployment administrative console:
   ```markdown
   http://dmgr_hostname:9060/ibm/console
   ```
   where `dmgr_hostname` is the fully qualified host name for the WebSphere Application Server Network Deployment.

5. Log into the deployment manager administrative console.

6. Increase the HTTP connection timeouts for the deployment manager.
   A. Click **System Administration > Deployment Manager > Web container transport chains**.
   B. Increase the timeout values. For the **WCInboundAdmin** and **WCInboundAdminSecure** entries listed in the web container transport chains section, complete the following steps to increase the timeout values:
      1. Click **HTTP Inbound Channel**.
      2. Change the **Read timeout** value to 180.
      3. Change the **Write timeout** value to 180.
      4. Save the configuration changes.

7. Change the timeout request period for the Java Management Extensions (JMX) connector.
   A. Click **System administration > Deployment Manager > Administration Services > JMX connectors > SOAPConnector > Custom Properties**.
   B. Select the **requestTimeout** property, and increase the value from 600 to 6000.
   C. Save the configuration changes.

8. Update the maximum Java heap size used by the deployment manager:
   A. Click **System administration > Deployment manager > Java and Process Management > Process Definition > Java Virtual Machine**.
B. Update the value in the **Maximum Heap Size** field. For information about appropriate heap sizes see the documentation for your operating system in the Performance Guides located on the WebSphere Portal and Web Content Management Product Documentation page. **Note:** If using a 32-bit operating system, you will need to set the heap size to a lower size than a 64-bit operating system.

C. Click **OK** and then save your changes.

9. Depending on your version of WebSphere Application Server, make the appropriate selection from the navigation:

   **Note:** If security is not enabled on your deployment manager, see [Enabling security](#) for information before performing this step.

   - For **WebSphere Application Server Version 6.1**: Click **Security > Secure administration, applications and infrastructure** and select **Enable Application Security**. Then save the configuration changes.

   - For **WebSphere Application Server Version 7.0**: Click **Security > Global security** and select **Enable Application Security**. Then save the configuration changes.

10. Verify that the WebSphere Portal administrative users and administrative group exist in the Deployment Manager cell's user registry. Perform the following steps if you need to create the administrative users and group:

   A. Click **Users and Groups > Manage Users**.

   B. Click **Create**.

   C. Type the information for the WebSphere Portal administrative users; for example **wpsadmin** and **wpsbind**, and then click **Create**.

   D. Click **Users and Groups > Manage Groups**.

   E. Click **Create**.

   F. Type **wpsadmins** as the name of the WebSphere Portal administrative group and then click **Create**.

   G. Click the group you just created; for example **wpsadmins**.

   H. Click the **Members** tab.

   I. Click **Add Users**.

   J. Search for the users.

   K. Select the users you want to add to the group.

   L. Click **Add** to add the users to the group.

   M. Click **Close** when you are done adding users to the group.

   N. Log out of the administrative console.

11. Change the timeout request period for the Simple Object Access Protocol (SOAP) client. Edit the **soap.client.props** file, located in the **Dmgr_profile\properties** directory: Change the line to: **com.ibm.SOAP.requestTimeout=6000**.

12. Run the following tasks to stop and restart the deployment manager:

   A. **stopManager.bat** -username **admin_userid** -password **admin_password**, from the **dmgr_profile_root\bin** directory

   B. **startManager.bat**, from the **dmgr_profile_root\bin** directory

---

Parent topic: [Preparing prerequisite and corequisite software on Windows](#)

Next topic: [Preparing a remote Web server when portal is installed on Windows](#)
Preparing a remote Web server when portal is installed on Windows

Install and configure the Web server plug-in, provided by IBM® WebSphere® Application Server, to set up your Web server to communicate with IBM WebSphere Portal.

- Prerequisites

  - Preparing the WebSphere Application Server Deployment Manager on Windows
  - Preparing prerequisite and corequisite software on Windows
  - Preparing your Windows operating system
  - Preparing the primary node on Windows

Perform the following steps to install and configure your Web server:

1. Install and configure the Web server; refer to the Web server documentation for information.

2. If using Microsoft Internet Information Server, we recommend updating the `UrlSegmentMaxLength` Registry key to a value of 0 to eliminate potential problems in a WebSphere Portal environment with the default IIS limitation on the length of URL path segments. We also recommend updating the `AllowRestrictedChars` Registry key to a value of 1 to accept hex-escaped characters in request URLs that decode to the U+0000 - U+001F and U+007F - U+009F ranges. **Note:** Refer to `Http.sys registry settings for IIS` for information.

3. If using IBM Lotus® Domino®, edit the `NOTES.INI` file on the Web server. Set the `HTTPEnableConnectorHeaders` parameter to 1.

4. If using IBM HTTP Server or Apache Server, edit the `httpd.conf` file on the Web server. Set the `AllowEncodedSlashes` directive to `on`; the directive should be added at the root level as a global directive.

5. Stop the Web server.

6. Install and configure the Web server plug-in on the system where the Web server is located using the Plug-ins installation wizard provided with WebSphere Application Server. Refer to `Selecting a Web server topology diagram and roadmap` for information. **Note:** The WebSphere Application Server Web server instructions are for the default profile. If not using the default profile, refer to `Configuring a Web server for a non-default profile`.

   **If using WebDAV:** After successfully installing the Web server plug-in, locate and open your `plugin-cfg.xml` file and set `AcceptAllContent` to `true`.

   **Important:** Depending on how you use the Web server, you may need to adjust the `ServerIOTimeout` value, which defines how long the plug-in should wait for a response from the application. The recommended minimum value is 60 but you may need to adjust this value higher if you are retrieving data from a database. To update this value, locate and open your `plugin-cfg.xml` file and set `ServerIOTimeout` to a value that is appropriate for your business needs. For additional information, see `Common questions about the Web server plug-in`.

7. If you are using Sun Web Server Version 7 update 8, read Technote 1448262 and perform the steps to resolve the HTTP 408/409 error.

8. Start the Web server.

**Parent topic:** Preparing prerequisite and corequisite software on Windows
Preparing your Windows operating system

View information on setting up your operating system for IBM® WebSphere® Portal. Other components might require additional steps; see the product documentation for the specific components you want to install for information.

- Prerequisites
  - Preparing prerequisite and corequisite software on Windows

Perform the following steps to prepare your operating system:

1. Check that the system logon user ID you will use during installation has the following permissions and rights:
   - The user ID must already exist prior to installation.
   - The user ID must belong to the Administrators group.

2. Perform the following steps to determine if a user account is a member of the Administrators group:
   A. Click Start > Programs > Administrative Tools > Computer Management.
   B. Expand Local Users and Groups and select Groups.
   C. Open the Administrators group to see what members belong to it.
   D. Add the user to the Administrators group if necessary.

3. Consider the following recommendations when installing to avoid excessively long path names: Note: If you exceed the 259 maximum character length, you may receive one of the following error messages during configuration or in the wpinstalllog.txt file:
   - The input line is too long.
   - The syntax of the command is incorrect.
   - The filename is too long.

   A. Use a short installation path. For example, use C:\WebSphere instead of C:\Program Files\IBM\WebSphere.
   B. Specify node names; do not use names longer than 5 characters. For example, you might use node1 instead of longnodename01.
   C. Name WAR files with less than 21 characters. If necessary, modify the file name before installing.

Parent topic: Setting up a cluster on Windows
Previous topic: Preparing prerequisite and corequisite software on Windows
Next topic: Preparing the primary node on Windows
Preparing the primary node on Windows

Before creating your high availability environment, you must install IBM® WebSphere® Portal on your primary node and then configure your database and your network deployment manager.

- **Prerequisites**
  - Preparing prerequisite and corequisite software on Windows
  - Preparing your Windows operating system

Perform the following tasks to prepare your primary node:

1. **Installing WebSphere Portal on Windows on the primary node**
   IBM WebSphere Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

2. **Configure WebSphere Portal to use a remote database**
   For high availability, using a remote database is preferred. For improved performance databases can be distributed across multiple database servers. Databases should also be configured for failover. Configuring the database for failover is beyond the scope of this documentation. Refer to the database product documentation for the most authoritative guidance about setting up databases for fail over.

3. **Configuring the primary node to communicate with the deployment manager on Windows**
   After installing IBM WebSphere Application Server Network Deployment and installing IBM WebSphere Portal on the primary node, you must configure the primary node and the deployment manager to communicate with each other.

4. **Removing search collections on Windows**
   If you plan to use search in a cluster, you must configure a remote search server; see Configuring Portal Search in a cluster for information. If you created any search collections, you will need to recreate them on the remote search server. If your search collection has data, export the collection before deleting it and then import the data on the remote server.

**Parent topic:** Setting up a cluster on Windows  
**Previous topic:** Preparing your Windows operating system  
**Next topic:** Choosing the type of cluster to create on Windows
Installing WebSphere Portal on Windows on the primary node

IBM® WebSphere® Portal is installed as a single component, complete with an integrated database for storing information. This allows you to get WebSphere Portal up and running quickly for a proof of concept phase where you can immediately begin building and working with it. You can also expand your environment to include high availability failover, a more robust database, and LDAP-based authentication.

Review the Supported hardware and software requirements for your operating system and ensure that all hardware and software matches the minimum product level before installing WebSphere Portal. Review Installation methods, options, and sources before installing to determine your installation source location and installation method.

Perform the following steps to install WebSphere Portal and create a custom profile for the node that will be added to the cluster. If you already installed WebSphere Portal, you can skip to the last step to create the custom profile.

1. Type `ping yourserver.yourcompany.com` on a command line to verify that your fully qualified host name is properly configured.
2. Type `ping localhost` on a command line to verify that your network is properly configured.
3. If you are installing with an existing WebSphere Application Server instance, ensure it is installed at the supported level and has the following features installed:
   - Application Server
   - Administration
     - Scripted Administration
     - Administrative Console
   - Ant and Deployment Tools
     - Deploy Tool
     - Ant Utilities
   - Important: Besides ensuring that the existing WebSphere Application Server instance is at the supported level, you will also need to ensure that Apache Derby is installed at the supported level before installing WebSphere Portal. See WebSphere Portal detailed system requirements for supported levels.

4. If you are installing on a server with a firewall, antivirus, and/or desktop search engine enabled, disable them before installing. If you do not disable them and the installation program detects them, a warning message displays during the installation.

5. Choose one of the following installation commands based on the installation method you want to use to install WebSphere Portal; see Installation Methods for more information:
   - **Advance installation parameters:** To customize your installation, you can add various parameters to your installation commands; for example, you can specify your port information. See Advanced installation parameters for information.
   - **Restriction:** When defining your installation path, the ONLY supported characters are ASCII letters [A-Z and a-z], numbers [0-9], slashes [/ or \, depending on your operating system], and the dash [-].

<table>
<thead>
<tr>
<th>Table 1. Installation task options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Installation type</strong></td>
</tr>
</tbody>
</table>

1993
6. Verify your installation was successful; access WebSphere Portal using the `http://yourserver:yourport/wps/portal` format. Run the `ConfigEngine.bat list-server-ports -DWasPassword=password` task from the `wp_profile_root\ConfigEngine` directory to generate the `wp_profile_root\ConfigEngine\log\wp_PortMatrix.txt` file that lists the WebSphere Portal ports for your installation.

7. Optional: After the WebSphere Portal installation completes successfully, run the `ConfigEngine.bat configure-express -DPortalAdminPwd=password -DWasPassword=password` task, from the `wp_profile_root\ConfigEngine` directory, if you want to enable the sample IBM Lotus Web Content Management content, such as internet and intranet sample sites.

   **Restriction:** Run the `configure-express` task before configuring your database, user registry, context root, security, etc. If you ran any tasks other than the install task, do not run this task.

   The sample content includes: **Note:** See [http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_ovw.html](http://publib.boulder.ibm.com/infocenter/wpexpdoc/v6r1m0/topic/com.ibm.wp.exp.doc/wcm/wcm_jumpstart_ovw.html) for tutorials on how to use the sample content.

   - Creates a group called `contentAuthors`; members of this group are given privileges to create content in the sample Internet and intranet sites. Navigate to the **Administration** area and then click **Access > Users and Groups**.
   - Creates two new Web Content Management Libraries: "Internet Web Content 6.1.0" and "Intranet Web Content 6.1.0". Navigate to the **Administration** area and then click **Portal Content > Web Content Libraries**.
   - Adds a portlet filter and applies the filter to various portlets in the sample Internet and intranet sites. You can see the definition of the filter in the WebSphere Application Server Administration console and examining the custom resources under the **Environment** area.
   - Creates two new theme policies: InternetStyle and IntranetStyle. These styles are applied to sample Internet and intranet sites. Navigate to **Theme Customizer** and then select the style.
   - Creates several portlet clones of the Web Content Management rendering portlet. These portlet clones are used on sample Internet and intranet sites.
   - Creates several sample credential slots, including "Default slot for E-mail", "Default slot for Feeds", "Default slot for Miscellaneous", "Default slot for Web Clipping", and "Default slot for Web Content Management". Navigate to the

---

**Note:** If the installation program does not detect a WebSphere Application Server instance that you know exists, exit the installation program and pass the WebSphere Application Server instance location using the command line; for example, `install.bat -W was.undetectedWas="/my/WAS/location"`. **Note:** If the GUI or console mode installation program fails to detect ports for either WebSphere Application Server or WebSphere Portal, a warning message displays and the installer offers another chance to enter the values. If the silent installation fails to detect ports for either WebSphere Application Server or WebSphere Portal, the installer will exit.

**Note:** To use the sample content, you need to create a group called `contentAuthors`; members of this group are given privileges to create content in the sample Internet and intranet sites. Navigate to the **Administration** area and then click **Access > Users and Groups**. Then, create two new Web Content Management Libraries: "Internet Web Content 6.1.0" and "Intranet Web Content 6.1.0". Navigate to the **Administration** area and then click **Portal Content > Web Content Libraries**. Add a portlet filter and apply the filter to various portlets in the sample Internet and intranet sites. You can see the definition of the filter in the WebSphere Application Server Administration console and examining the custom resources under the **Environment** area. Create two new theme policies: InternetStyle and IntranetStyle. These styles are applied to sample Internet and intranet sites. Navigate to **Theme Customizer** and then select the style. Create several portlet clones of the Web Content Management rendering portlet. These portlet clones are used on sample Internet and intranet sites. Create two virtual portals with context roots of `wps/portal/intranet` and `wps/portal/internet`. These are the sample Internet and intranet sites. Go to [http://yourserver:yourport/wps/portal/internet](http://yourserver:yourport/wps/portal/internet) and [http://yourserver:yourport/wps/portal/intranet](http://yourserver:yourport/wps/portal/intranet) to access them. Create several sample credential slots, including "Default slot for E-mail", "Default slot for Feeds", "Default slot for Miscellaneous", "Default slot for Web Clipping", and "Default slot for Web Content Management". Navigate to the...
Administration area and then click Access > Credential Vault > Manage System Vault Slots.
- Modifies the portlet palette to include some new portlets. Click the Portlets link to see the portlet palette.

8. Optional: If you ran the configure-express task, the owner of the items in the Web content libraries containing the Internet and Intranet Site Template content will be listed as uid=xyzadmin,o=defaultWIMFileBasedRealm. To update the owner information for these items to correspond to your portal administrator ID, complete the following steps:

A. Edit the wp_profile_root\PortalServer\wcm\shared\app\config\wcmservices\MemberFixerModule.properties file.
B. Add the following line to the file:
   
   `uid=xyzadmin,o=defaultWIMFileBasedRealm -> portal_admin_DN`
   
   Replace `portal_admin_DN` with the distinguished name of the portal administrator.
C. Save your changes and close the file.
D. Run the ConfigEngine.bat action-express-memberfixer -DPortalAdminPwd=yourpassword -DWasPassword=yourpassword task, located in the wp_profile_root\ConfigEngine directory.

9. Optional: Perform the following steps if you need to change the ports for WebSphere Application Server or WebSphere Portal:

   **Note:** The starting port parameter is required for a successful completion of the modify-ports-by-startport task.

   Once you specify a start port, this port becomes the base for assigning port values. The code increments this value as each port is assigned, which means that the WebSphere Portal ports will be assigned incrementally starting with the port defined with the modify-ports-by-startport task.

   A. Stop the server1 and WebSphere_Portal servers.
   B. Run one of the following commands for each server you need to change:

<table>
<thead>
<tr>
<th>Method</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting port number</td>
<td>ConfigEngine.bat modify-ports-by-startport -DWasPassword=password -DModifyPortsServer=servername -DStartPort=starting port number</td>
</tr>
</tbody>
</table>

   The following is an example of the information within a port file although the port values will be different based on your environment:

   ```
   BOOTSTRAP_ADDRESS=10031
   SOAP_CONNECTOR_ADDRESS=10033
   SIB_MQ_ENDPOINT_ADDRESS=10030
   SIB_MQ_ENDPOINT_SECURE_ADDRESS=10037
   SIB_ENDPOINT_ADDRESS=10026
   SIB_ENDPOINT_SECURE_ADDRESS=10037
   ```

C. Restart the server1 and WebSphere_Portal servers.

**Parent topic:** Preparing the primary node on Windows

**Next topic:** Configure WebSphere Portal to use a remote database
Related concepts
Installation methods
IBM Support Assistant Lite for WebSphere Portal

Related reference
Advanced installation parameters
Configure WebSphere Portal to use a remote database

For high availability, using a remote database is preferred. For improved performance databases can be distributed across multiple database servers. Databases should also be configured for failover. Configuring the database for failover is beyond the scope of this documentation. Refer to the database product documentation for the most authoritative guidance about setting up databases for fail over.

Password considerations when transferring data manually

For security reasons, you should not store passwords in the wkplc.properties and wkplc_comp.properties. Edit each of the properties files prior to running a configuration task, inserting the passwords needed for that task. After the task has run, delete all passwords from each file. For more information, see Deleting passwords from configuration scripts. Alternatively, you can specify the password on the command line using the following syntax:

- Windows: ConfigEngine.bat task_name -Dpassword_property_key=password_value
- UNIX: ./ConfigEngine.sh task_name -Dpassword_property_key=password_value
- i5/OS: ConfigEngine.sh task_name -Dpassword_property_key=password_value

As with other properties, each password property must have the -D prefix and be set equal to (=) a value. If you have multiple properties in a single command, use a space character between each -Dproperty=value setting.

- Prerequisites

  - Installing WebSphere Portal on Windows on the primary node
  - Technotes for database connectivity issues

- Configuring WebSphere Portal to use DB2

  View information on installing and setting up DB2 to work with WebSphere Portal.

- Configuring WebSphere Portal to use DB2 for z/OS

  View information on installing and setting up DB2 for z/OS to work with WebSphere Portal.

- Configuring WebSphere Portal to use Oracle

  View information on installing and setting up Oracle to work with WebSphere Portal.

- Configuring WebSphere Portal to use Oracle RAC

  View information on installing and setting up Oracle RAC to work with WebSphere Portal.

- Configuring WebSphere Portal to use SQL Server

  View information on installing and setting up SQL Server to work with WebSphere Portal.

- Verifying databases

  After you configure IBM WebSphere Portal to work with your database, test the database connection to ensure that it operates correctly. Then verify that all database transactions work properly within the WebSphere Portal environment. For example, all portal pages should display without HTTP 404 errors, and there should be no database layer-related exceptions in the SystemOut.log and SystemErr.log files.

Parent topic: Preparing the primary node on Windows

Previous topic: Installing WebSphere Portal on Windows on the primary node

Next topic: Configuring the primary node to communicate with the deployment manager on Windows

Related Information
Troubleshooting WebSphere Portal Version 6.1 databases
Configuring WebSphere Portal to use DB2

View information on installing and setting up DB2 to work with WebSphere Portal.

1. **Installing DB2**
   View information on installing DB2 for use with WebSphere Portal.

2. **Create users**
   View the steps to set up users for DB2 to work with WebSphere Portal.

3. **Creating remote databases**
   A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

4. **Optional: Assigning custom table spaces**
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

5. **Modifying database properties**
   Learn how to modify the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

6. **Optional: Configuring JCR collation support**
   View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

7. **Configuring WebSphere Portal to use DB2**
   View information on manually transferring data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

8. **Configuring DB2 for large file handling in Web Content Management**
   If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

9. **Optional: Changing driver types**
   WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

**Parent topic:** Configure WebSphere Portal to use a remote database
Installing DB2
View information on installing DB2 for use with WebSphere Portal.

Before you begin:
- **(For Type 2 drivers only):** DB2 Connect must be installed on the same system as WebSphere Portal, and have the same name as the server profile name.
- When you install DB2 using the DB2 installation program, it automatically creates a DB2 administrative user with the correct operating system rights.

1. If you are using the JDBC driver in type 2 mode, configure your DB2 Connect client with the following commands. If you are using a remote database, complete this step separately from the WebSphere Portal installation.
   - `db2 update dbm cfg using tp_mon_name WAS`
   - `db2 update dbm cfg using spm_name hostname`, where `hostname` is the host name of WebSphere Portal. Because the default for `spm_name` is the hostname itself, specifying the `hostname` parameter is optional. If your hostname is more than eight characters, use empty double quotes (" "). For example, `db2 update dbm cfg using spm_name " "`.

2. Before installing DB2, log in with a user ID that has administrative authority. This user should have the following specifications:
   - Belong to the local Administrator group
   - Act as part of the operating system
   - Have permissions to create a token object
   - Windows 2003 only: Have permissions to adjust memory quotas for a process
   - Have permissions to replace a process level token

   To edit user rights:
   - For the first two specifications: Click **Start > Programs > Administrative Tools > Computer Management > Local Users and Groups**.
   - For the last four specifications: Click **Start > Programs > Administrative Tools > Local Security Policy**. Then, click **Local Policies > User Rights Assignment**.

3. To install DB2, follow the instructions that are provided with the DB2 documentation to install DB2 and the required fix pack.

Parent topic: Configuring WebSphere Portal to use DB2
Next topic: Create users

Related Information
- **DB2 Technical Support**
- **DB2 Information Center**
Create users

View the steps to set up users for DB2 to work with WebSphere Portal.

Before you begin: You should have completed Installing DB2.

- **Prerequisites**
  - Installing DB2

For all databases, use one user with administrative rights on the operating system and the DB2 installation. This user can be the database administrative user that is created automatically by the DB2 installation program. This user is the database configuration user that will be used for configuration tasks: creating database tables and performing database transfer. If you choose to have WebSphere Portal also create the databases, the database user should be given SYSADM rights. You only need to manually create an administrator ID when you do not want to use an existing DB2 administrator ID.

A common user name is `db2inst1`, but you can assign any user name as long as it has administrative access and follows the limitations listed here. Do not change the user name after creating it.

The limitations on user names are:

- User names can contain one to 20 characters.
- Group and instance names can contain one to eight characters.
- Names cannot be any of the following: `users`, `admins`, `guests`, `public`, `local`.
- Names cannot begin with: `IBM`, `SQL`, `SYS`.
- Names cannot include accented characters.
- Create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Following are the basic steps to create new users and grant privileges. For more information, refer to the DB2 product documentation.

1. From the DB2 Control Center, expand the object tree until you find the DB Users folder.
2. Right-click the DB Users folder and select **Create/Add**. The Create DB User notebook opens.
3. In the User name field, type a user ID.
4. Assign an authority for the user.
5. Click **OK**.

**Parent topic:** Configuring WebSphere Portal to use DB2  
**Previous topic:** Installing DB2  
**Next topic:** Creating remote databases
Related Information

DB2 product documentation
Creating remote databases

A remote database resides on a different system than WebSphere Portal. When you use a remote server, you must manually create the databases that are required by WebSphere Portal.

Before creating these databases, note the following information:

- Use IBM® DB2 Universal Database™ Enterprise Server Edition JDBC driver, JCC, type 4.
- If the DB2 JDBC type 4 driver is used, only the instructions on the remote server are required. You do not need to install the DB2 client software, and you do not need to complete the steps related to the DB2 client.
- The client software, DB2 Connect, must be correctly configured to connect to the remote DB2 server instance, for example, `db2inst1`.
- These instructions assume that a remote DB2 server and DB2 Connect are already installed and running. For information on installing DB2 Connect, see the DB2 product documentation.

**Prerequisites**
- Installing DB2
- Create users

To create a database, you must be a DB2 System Administrator with sufficient database privileges (SYSADM or at a minimum SYSCtrl).

1. Log in to the DB2 server system as a user with sufficient database privileges.
2. Ensure that the database user has been created, granted appropriate privileges, and has a password assigned to it. If the user has not been created, refer to the Creating users topic for information on how to create users.
3. Initialize a DB2 command environment by opening a command prompt and typing `db2cmd`. In this mode, you must type `db2` at the beginning of each command and use double quotation marks (`"`) around the entire command. The following example shows how to use the double quotes; it is not an actual command that you run:

   ```
   db2 "CREATE REGULAR TABLESPACE SMS PAGESIZE 4 K MANAGED BY SYSTEM USING ('sms') BUFFERPOOL SMSPOOL"
   ```

   **Note:** If you are using the Command Line Processor (CLP), refer to your DB2 documentation for details. The command prompt is `db2->`. In this mode, commands can be entered without the `db2` prefix or the double quotation marks.

   However, the following steps assume you are not using the CLP and are entering commands from the `$` prompt.

4. Run the following commands on the DB2 server system to configure the DB2 database instance:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Description</th>
</tr>
</thead>
</table>
| DB2         | `db2set DB2_RR_TO_RS=YES`  
|             | `db2set DB2_EVALUNCOMMITTED=YES`  
|             | `db2set DB2_INLIST_TO_NLJN=YES`  
|             | `db2 "UPDATE DBM CFG USING query_heap_sz 32768"`  
|             | `db2 "UPDATE DBM CFG USING maxagents 500"`  
|             | `db2 "UPDATE DBM CFG USING sheapths 0"`  

5. **(Important)** Failure to complete this step will result in an unsuccessful database transfer. Run the following commands on the DB2 server system to create the necessary databases:

   - Replace `dbname` with the actual name of the database. Run the commands and each time replace `dbname` with the actual values for release, community, customization, Java Content Repository, Feedback, and Likeminds. You will need to run the commands once for each database for a total of six times.

   **Remember:** DB2 database names cannot
exceed eight characters. Therefore, consider using these database names: release, commun, custom, jcrdb, fdbkdb, and lmdb.

```
6. Complete the following:

A. On the DB2 server system, run the following commands. This step is only required for the IBM Java Content Repository database (jcrdb).

- jcrdb is the name of the database used to store user data and objects
- jcr is the jcr user for jcrdb
- dbpassword is the password for the jcr user for the jcrdb

```
```
db2 "CREATE DB dbname using codeset UTF-8 territory us PAGESIZE 8192"
db2 "UPDATE DB CFG FOR dbname USING applheapsz 4096"
db2 "UPDATE DB CFG FOR dbname USING app_ctl_heap_sz 1024"
db2 "UPDATE DB CFG FOR dbname USING stmheap 32768"
db2 "UPDATE DB CFG FOR dbname USING dbheap 2400"
db2 "UPDATE DB CFG FOR dbname USING locklist 1000"
db2 "UPDATE DB CFG FOR dbname USING logfilsiz 4000"
db2 "UPDATE DB CFG FOR dbname USING logprimary 12"
db2 "UPDATE DB CFG FOR dbname USING logsecond 20"
db2 "UPDATE DB CFG FOR dbname USING logbufsz 32"
db2 "UPDATE DB CFG FOR dbname USING avg_appls 5"
db2 "UPDATE DB CFG FOR dbname using AUTO_MAINT off"
```
```
B. On the DB2 server system, check the services file. If it does not specify DB2 connection and interrupt service ports, specify the ports for your operating system: Use a text editor to open the file /etc/services and add the following text:
```
db2c_db2inst1port1/tcp
```
where db2inst1 is the default instance and port1 is the TCP port DB2 listens on. Replace port1 with the actual port
number that assigned to the DB2 connection service in your DB2 server installation.

The /etc/services file is located under %systemroot%/system32/drivers/, where %systemroot% is the location of the operating system. For example C:\Windows\system32\drivers\etc\services.

C. For JDBC Type 2 connections only: On DB2 Connect, use a text editor to open the /etc/services file. If it does not specify the DB2 connection service port, add the following text to specify the port for the remote DB2 instance:

db2c_db2inst1 port1 # DB2 connection service port

where db2inst1 is the name of the DB2 instance ID on the system, and port1 with the actual port number that is assigned to the DB2 connection service in your DB2 server installation. The connection service port on the DB2 Client system, WebSphere Portal server, must match the connection service port on the DB2 server. The ports should match by number but not necessarily by name.

D. Set up the correct service name by entering the following command on the DB2 server system:

db2 "UPDATE DBM CFG USING svcename svce_name" where svce_name is the connection service port name that is specified in substep b, such as db2c_db2inst1.

E. For JDBC Type 2 connections only: On DB2 Connect, set DB2COMM to TCP/IP by using the db2set command:

db2set DB2COMM=tcpip.

F. For JDBC Type 2 connections only: Catalog the TCP/IP node with the IP address of the remote database server on DB2 Connect:

db2 "catalog tcpip node remote_db_node_alias remote database_server_node_name connection_service_port" where:

- remote_db_node_alias is the alias name of the database server that you are defining for the WebSphere Application Server node name. The alias name can contain one to eight characters.
- database_server_node_name is the fully qualified host name of your database server system.
- connection_service_port is the name of the DB2 connection service port that is configured in the /etc/services file on the database server system.

G. For JDBC Type 2 connections only, catalog the WebSphere Portal databases on DB2 Connect, where:

- remote_db_name_domain, is the cataloged name of the databases on the server system for each domain.
- domain_alias_name, is the database alias names that you are defining.
- remote_db_node_alias is the name that was used previously when you cataloged the TCP/IP node in the previous step.

The alias for each database must be different from the actual database name and can only contain up to eight characters.

db2 "catalog db remote_db_name_release as release_alias_name at node remote_db_node_alias"
db2 "catalog db remote_db_name_community as comm_alias_name at node remote_db_node_alias"
db2 "catalog db remote_db_name_customization as cust_alias_name at node remote_db_node_alias"
db2 "catalog db remote_db_name_fdbkdb as fdbkdb_alias_name at node remote_db_node_alias"
db2 "catalog db remote_db_name_lmdb as lmdb_alias_name at node remote_db_node_alias"
db2 "catalog db remote_db_name_jcrdb as jcrdb_alias_name at node remote_db_node_alias"

H. For JDBC Type 2 connections only: Log out of DB2 Connect by entering db2 "terminate".

I. For JDBC Type 2 connections only, on DB2 Connect, test your remote connection by issuing the following command in the DB2 command window:

db2 "connect to alias_name user username using password", where alias_name is the alias name that you defined above, username is the database user, and password is the password assigned to the database user.
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:

- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- The page size of table spaces used by WebSphere Portal must be 8192 bytes.
- For details on creating table spaces refer to the documentation for the database.

**Prerequisites**

- Installing DB2
- Create users
- Creating remote databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file `wp_profile_root`/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - `dbdomain.table_name.tablespace`
   - `dbdomain.table_name.index_name.indexspace`
   
   For the file name and each table space and index space property pair, `dbdomain` can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds

3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword `IN` and a space. For example: `community.COMP_INST.tablespace=IN COMM8KSPACE`
   
   Repeat this step for each domain that you are transferring.

4. Save and close `dbdomain.space_mapping.properties`.
5. From a command prompt, specify the option `-DuseCustomTablespaceMapping=true` when starting the database transfer. For example,
   - Windows: `ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true`
   - UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

**Parent topic:** Configuring WebSphere Portal to use DB2  
**Previous topic:** Creating remote databases  
**Next topic:** Modifying database properties
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data. Create a backup copy of each properties file before modifying it.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**

- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:
  - release(DbName)=release
  - jcr(DbName)=jcrdb
  - feedback(DbName)=fdbkdb
  - likeminds(DbName)=lmdb
  - community(DbName)=commdb
  - customization(DbName)=custdb

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds

- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment. Note: The wkplc_comp.properties file by default has the properties populated for the Apache Derby database. The source.dbdomain.properties represent the properties for the source database. These values should not be modified unless you are transferring from a supported database other than Apache Derby.

   A. For dbdomain.DbType, type db2.
   B. For dbdomain.DbName, type the name of the WebSphere Portal domain database. Note: This value is also the database element in the dbdomain.DbUrl property.
   C. For dbdomain.DbName, type the name of the WebSphere Portal domain database. DB2 database names cannot exceed eight (8) characters. Note: This value is also the database element in the dbdomain.DbUrl property.
   D. For dbdomain.DbSchema, type the schema name of the database domain. Note: Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand.
   E. For dbdomain.DataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback
   F. For dbdomain.DbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. Note: The database element of this value should match the value of DbName.
   G. For dbdomain.DbUser, type the user ID for the database administrator.
   H. For dbdomain.DbPassword, type the password for the database administrator.
   I. Optional: For dbdomain.DbRuntimeUser, type the user ID of the database user that should be used by WebSphere Portal to connect to the database at runtime.
   J. If dbdomain.DbRuntimeUser is specified, you must set dbdomain.DbRuntimePassword to be the password of the runtime database user.
   K. For dbdomain.DBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.
L. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.

   A. For `source.domain.DbType`, type of the database you are currently configured to use. The value for `source.domain(DbType)` is Derby by default.

   B. For `source.domain.DbName`, type the name of the database domain you are currently using.

   C. For `source.domain.DbSchema`, type current schema identifier for objects within the database for this domain.

   D. For `source.domain.DataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.

   E. For `source.domain.DbUrl`, type the url currently used to access your database.

   F. For `source.domain.DbUser`, type the name of the user accessing this database.

   G. For `source.domain.DbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.

   A. For `db2.DbDriver`, type the name of the JDBC driver class.

   B. For `db2.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.

   C. For `db2.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.

   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Configuring WebSphere Portal to use DB2  
**Previous topic:** Assigning custom table spaces  
**Next topic:** Configuring JCR collation support
Configuring JCR collation support

View the steps to set up JCR collation to work with your IBM® DB2 Universal Database™ Enterprise Server Edition database. JCR collation is recommended when the language locales of your users do not natively collate correctly in the DB2 database and when language locale correct ordering is important.

- **Prerequisites**
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties

1. **Stop the WebSphere Portal server:**
   ```
   stopServer.bat WebSphere_Portal -username admin_userid -password admin_password
   ```

2. **If using a remote DB2 server**, copy the following files from the WebSphere Portal server to a temporary directory on the remote DB2 server:
   ```
   PortalServer_root/jcr/prereq.jcr/config/collation.jar
   PortalServer_root/jcr/prereq.jcr/config/registerCollationUDFTemplate.sql
   ```

3. **Set up collation on the database where the JCR domain is located.** Log in to the database machine with a userid that is authorized and configured to use the appropriate DB2 instance. For example, a common userid is `db2inst1`.
   A. Change to the directory `db2home/function`.
   B. Execute the command:
      ```
      Remote DB2: db2home/java/jdk/bin/jar -xvf temporary location/collation.jar
      Local DB2: db2home/java/jdk/bin/jar -xvf PortalServer_root/jcr/prereq.jcr/config/collation.jar
      ```
   C. Change to the directory `wp_profile_root\PortalServer\jcr\config`. If using a remote DB2, change to the temporary directory where you copied the files in a previous step.
   D. Open the file `registerCollationUDFTemplate.sql` and change all `SCHEMA` references to the JCR schema; for example, `JCR`. The value set for `SCHEMA` should match the value set for the `jcr.DbSchema` property.
   E. Connect to the JCR database by running `db2 connect to <jcrdb> user <userid> using <password>`.
   F. Execute the script by running the command:
      ```
      Remote DB2: db2 -tvf temporary location/registerCollationUDFTemplate.sql
      Local DB2: db2 -tvf wp_profile_root\PortalServer\jcr\config/registerCollationUDFTemplate.sql
      ```
   G. Disconnect from the JCR database.
   H. Restart the DB2 instance.

4. **Verify that the UDF is registered properly.**
   A. Log in as the `db2instanceID`. Open a DB2 terminal window, and type `db2`. From the command line, type the command `connect to <jcrdb> user <userid> using <password>`. You must connect to the JCR database using the userid and password of the WebSphere Portal JCR data source.
   B. Execute the following command in the previous substep. If the command completes successfully, the UDF is registered correctly as follows: `values schema.sortkeyj('abc','en')`, where `schema` is the schema used in the previous substep.

5. **Edit the `icm.properties` file**, located in `wp_profile_root\PortalServer\jcr\lib\com\ibm\icm` directory. Add the following section to the end of the file:
   ```
   # Enable/Disable collation support for all DB2 platforms
   ```
# Disabled by default

jcr.query.collation.db2.enabled = true

# Database specific collation mappings

# These mappings apply map a Java locale name into a collation name
# supported by the underlying database.
# Example mappings for DB2 platform

# English
jcr.query.collation.en = en

# Swedish
jcr.query.collation.sv = sv

jcr.query.collation.zh = zh
jcr.query.collation.de = de
jcr.query.collation.da = da
jcr.query.collation.hu = hu
jcr.query.collation.jp = jp

6. Enter the following command to start the WebSphere Portal server:

```
startServer.bat WebSphere_Portal
```
Configuring WebSphere Portal to use DB2

View information on manually transferring data to the DB2 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

- Ensure that the following prerequisites are met:
  - Supported database software is installed.
  - Databases and users are set up.

**Prerequisites**
- Installing DB2
- Create users
- Creating remote databases
- Modifying database properties

**Tips:**

- To run these tasks as a non-root user, you must first run the task chown -R non-root_userWebSphereDir.
- If you are transferring from Oracle, the open_cursors setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.
- Be sure that DB2 is started by checking the service. If attempts to restart result in a logon failure message, then go to the DB2 properties and reenter the password.

**Steps for transferring data to another supported database**

1. If you are running a type 2 connection, before transferring data edit the db2cli.ini file. Failure to follow these steps will cause the database transfer to hang at the task action-process-constraints.
   
   A. Locate the file C:\Program Files\IBM\SQLLIB\db2cli.ini.
   
   B. Add the following to the end of the file. Leave an empty line after ReturnAliases=0.[COMMON]

   ```
   DYNAMIC=1
   ReturnAliases=0
   ```

2. Open a command prompt and change to the directory wp_profile_root\ConfigEngine.

3. Enter the following commands to validate configuration properties. ConfigEngine.bat validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password

   ```
   ConfigEngine.bat validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```

4. From the same command prompt as the previous steps, change to the directory wp_profile_root\bin.
5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td><code>stopServer.bat server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td><code>stopServer.bat WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>

6. Transfer the database. **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

A. Change to the directory `wp_profile_root\ConfigEngine`.

B. Enter the following command:

```bash
ConfigEngine.bat database-transfer -DWasPassword=password
```

**Note:**
- To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:

```
ConfigEngine.bat database-transfer -DTransferDomainList=jcr -DWasPassword=password
```
- If you have been storing data in IBM® Derby for a long time, database transfer could fail with OutOfMemory exceptions. If database transfer fails, add the following property to the command in this step:

```
ConfigEngine.bat database-transfer -DDbtJavaMaxMemory=1536M -DWasPassword=password
```

C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

7. After transferring the database tables, perform a reorg check to improve performance. Perform this step for each database alias in the property file.

A. Connect to a database with the following command:

```
db2 connect to database_alias user db2admin_userid using password
```

**Note:** Additional options might be required if additional security has been installed. Refer to DB2 Universal Database commands by example for links to the command reference.

B. After it is connected, run the following command from the DB2 prompt:

```
reorgchk update statistics on table all > xyz.out
```

C. Look in the reorg column for entries marked with a * (star or asterisk) in the file `xyz.out`. For each line with a *, note the `tablename` and run the following command for each `tablename`:

```
db2 terminate
db2rbind database_name -l db2rbind.out -u db2_admin -p password
```

D. The output file `db2rbind.out` is only created when there is an error for the `db2rbind` command.

8. Change to the directory `wp_profile_root\bin`.

9. Enter the following command to start the WebSphere Portal server:

```
startServer.bat WebSphere_Portal
```

**Parent topic:** Configuring WebSphere Portal to use DB2  
**Previous topic:** Configuring JCR collation support  
**Next topic:** Configuring DB2 for large file handling in Web Content Management

**Related Information**
Configuring DB2 for large file handling in Web Content Management

If you are using Web Content Management, you must update the database configuration to support large files. Do this by setting the `fullyMaterializeLobData` property in the WebSphere Application Server administrative console.

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties
  - Configuring WebSphere Portal to use DB2

Note: You only need to perform these steps if you are using Web Content Management.

1. Log into the WebSphere Application Server administrative console.
2. Click **Resources** > **JDBC** > **Data sources**.
3. Select all scopes (the default setting) or select a specific cell, node, or node/server. Select the scope that corresponds to your instance of WebSphere Portal. The view refreshes.
4. Select the name of the data source that is defined in `wkplc_comp.properties` for the JCR database domain. The default data source is `wpdbDS`.
5. Click **Custom properties**.
6. Ensure that the `fullyMaterializeLobData` property is set to `false`.

Parent topic: Configuring WebSphere Portal to use DB2
Previous topic: Configuring WebSphere Portal to use DB2
Next topic: Changing driver types
Changing driver types

WebSphere Portal requires either JDBC Type 2 or Type 4 drivers. View the steps to switch from one driver type to the other.

Before you begin, ensure that the following conditions are met:
- The WebSphere Portal database has been successfully transferred to DB2 using the `database-transfer` configuration task.
- The files `wkplc_comp.properties` and `wkplc_dbtype.properties` have been modified to set the correct values for the DB2 drivers that you are switching to: In the file `wkplc_comp.properties` set each `<Domains>.DbUrl` property using the following formats:

  ```
  # db2 (type 2):        { jdbc:db2:wpsdb } 
  # db2 (type 4):        { jdbc:db2://<YourDatabaseServer>:50000/wpsdb:returnAlias=0; } 
  ```

  In the file `wkplc_dbtype.properties` set the `db2.DbLibrary` property using the following format: For DB2 Type 2 driver use `<SQLLIB>/java/db2java.zip` 

  For DB2 Type 4 driver use `<SQLLIB>/java/db2jcc.jar;<SQLLIB>/java/db2jcc_license_cu.jar`

  In the file `wkplc_dbtype.properties` set the `db2.DbDriver` property using the following format: For DB2 Type 2 driver use `COM.ibm.db2.jdbc.app.DB2Driver`

  For DB2 Type 4 driver use `com.ibm.db2.jcc.DB2Driver`

- Prerequisites
  - Installing DB2
  - Create users
  - Creating remote databases
  - Modifying database properties
  - Configuring WebSphere Portal to use DB2
  - Configuring DB2 for large file handling in Web Content Management

1. Open a command prompt and change to the directory `wp_profile_root\ConfigEngine`

2. Enter the following commands to validate configuration properties. `ConfigEngine.bat validate-database-driver` 

   ```
   DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password 
   ```

3. From the same command prompt as the previous steps, change to the directory `wp_profile_root\bin`.

4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebSphere Application Server</strong></td>
<td>stopServer.bat server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td><strong>WebSphere Portal</strong></td>
<td>stopServer.bat WebSphere_Portal -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

5. Change to the directory `wp_profile_root\ConfigEngine`.

6. To change from one supported driver to the other, run the following task to connect the database, including only the domains that require the switch: `ConfigEngine.bat connect-database`
7. Change to the directory `wp_profile_root\bin`.
8. Enter the following command to start the WebSphere Portal server:
   `startServer.bat WebSphere_Portal`
Configuring WebSphere Portal to use DB2 for z/OS

View information on installing and setting up DB2 for z/OS to work with WebSphere Portal.

1. Installing DB2 for z/OS
   View information on how to install DB2 for z/OS for use with IBM® WebSphere® Portal.

2. Creating users
   Complete this procedure to set up users for IBM DB2 Universal Database™ for z/OS® to work with IBM WebSphere Portal.

3. Creating remote databases
   A remote database resides on a different machine than WebSphere Portal. You must manually create the required databases before configuring WebSphere Portal to work with DB2 for z/OS.

4. Optional: Creating the Java Content Repository database
   View information on setting up your Java Content Repository database to work with WebSphere Portal.

5. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

6. Configuring WebSphere Portal to use DB2 for z/OS
   View information on manually transferring data to the DB2 for z/OS you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2 for z/OS. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configure WebSphere Portal to use a remote database
Installing DB2 for z/OS

View information on how to install DB2 for z/OS for use with IBM® WebSphere® Portal.

The following prerequisites must exist separately from the WebSphere Portal installation:

- If you are using the DB2 Legacy Type 2 JDBC driver, configure your DB2 Connect client with the following commands:
  - `db2 update dbm cfg using tp_mon_name WAS`
  - `db2 update dbm cfg using spm_name hostname`, where `hostname` is the host name for the machine where the DB2 Connect client is installed (same as portal machine).

- If you are using the DB2 Legacy Type 2 JDBC driver, enable extended shared memory usage with the following commands:
  - `export EXTSHM=ON`
  - `db2set DB2ENVLIST=EXTSHM`
  - `db2start`

To make the change permanent, you must add the environment variable by adding the following to the `profile.env` file:

```
DB2ENVLIST='EXTSHM'
```

in `/home/db2inst/sqllib/userprofile`

add: `export EXTSHM=ON`

**Note:** The shell must be reopened before restarting DB2 for z/OS.

- Both type 2 and type 4 drivers are supported. If you are using the DB2 Legacy Type 2 JDBC driver, the DB2 Connect client must be installed on the same machine as WebSphere Portal to connect to the remote database.
- The DB2 subsystem must be on a supported z/OS platform.
- Update the BP8K0 catalog bufferpool to 35,000 buffers before performing database transfer. You should change this value based on your environment. The SYSIBM.SYSDATABASE table resides in this bufferpool and is used extensively by DB2 for z/OS during database transfer.

To use DB2 for z/OS as the database software for WebSphere Portal, you must have DB2 for z/OS installed on your z/OS system. Refer to the DB2 for z/OS product documentation for instructions. Refer to the Planning for DB2 for z/OS topic for additional considerations for configuring DB2 for z/OS as the WebSphere Portal database.

**Parent topic:** Configuring WebSphere Portal to use DB2 for z/OS

**Next topic:** Creating users
Creating users

Complete this procedure to set up users for IBM® DB2 Universal Database™ for z/OS® to work with IBM WebSphere® Portal.

Before you begin: You should have completed Installing DB2 for z/OS.

- Prerequisites
  - Installing DB2 for z/OS

- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.
- If you are configuring multiple WebSphere Portal instances to use a single DB2 for z/OS subsystem, you must create a unique database user for each WebSphere Portal instance. By default all database table names include the name of the database user used to access the data. Therefore, to prevent table name conflicts, you must create a unique database user for each WebSphere Portal instance on the shared DB2 for z/OS subsystem.
- Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

Use the following steps to grant access permissions to the database users. Repeat these steps for each WebSphere Portal instance you are setting up.

1. Create all database user IDs in the security product you are using on z/OS. For jcrschema, the database schema name for IBM Java Content Repository data, create a group and connect it to the database user ID for Java Content Repository data, jcr. The following sample shows the RACF definition of such a user ID and group, where jcr is the database user ID for Java Content Repository data, yourDefaultUserGroup is your default RACF group for database user IDs, jcrschema is the database schema name for Java Content Repository data, and yourDefaultGroup is your default RACF group for groups. If you have some other security product such as Top Secret or ACF2 instead of RACF, translate the sample RACF definition into the appropriate syntax before running the job.

   ADDUSER jcr DFLTGRP(yourDefaultUserGroup) NAME('WAS DB2 ACCESS USER')
   PW USER(jcr) NOINTERVAL
   ALU jcr PASSWORD(********) NOEXPIRED
   ADDGROUP jcrschema SUPGROUP(yourDefaultGroup)
   CONNECT jcr GROUP(jcrschema)

2. Run the following SQL statements using a tool like SPUFI while logged on to the DB2 subsystem to grant appropriate rights on the newly created databases. If you are configuring multiple WebSphere Portal instances to use a single DB2 for z/OS subsystem, be sure to use the database user associated with the database user ID for Java Content Repository data, jcr. The following sample shows the SQL statements for granting access rights.

   *Note:* Make sure your vmmdb user follows these same standards.

   WebSphere Portal instance you are setting up. (C) create/alter tablespaces
   (C) create/alter tables
   (C) create/alter indice;
   (C+R) read/write data
   (C) - at configuration time
   (R) - at runtime
GRANT USE OF ALL BUFFERPOOLS TO releaseusr;
GRANT USE OF ALL BUFFERPOOLS TO communityusr;
GRANT USE OF ALL BUFFERPOOLS TO customizationusr;

GRANT DBADM ON DATABASE jcrdbnameonzos TO jcr;
GRANT USE OF ALL BUFFERPOOLS TO jcr;
GRANT DBADM ON DATABASE fdbkdbnameonzos TO feedback;
GRANT USE OF ALL BUFFERPOOLS TO feedback;
GRANT DBADM ON DATABASE lmdbnameonzos TO lmdbusr;
GRANT USE OF ALL BUFFERPOOLS TO lmdbusr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO releaseusr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO communityusr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO customizationusr;

GRANT SELECT ON SYSIBM.SYSTABLES TO releaseusr;
GRANT SELECT ON SYSIBM.SYSTABLES TO communityusr;
GRANT SELECT ON SYSIBM.SYSTABLES TO customizationusr;

GRANT SELECT ON SYSIBM.SYSCOLUMNS TO jcr;
GRANT SELECT ON SYSIBM.SYSTABLES TO jcr;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO fdbkdbus;
GRANT SELECT ON SYSIBM.SYSTABLES TO fdbkdbus;
GRANT SELECT ON SYSIBM.SYSCOLUMNS TO lmdbusr;
GRANT SELECT ON SYSIBM.SYSTABLES TO lmdbusr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO releaseusr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO communityusr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO customizationusr;

GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO jcr;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO fdbkdbus;
GRANT SELECT ON SYSIBM.SYSFOREIGNKEYS TO lmdbusr;
GRANT SELECT ON SYSIBM.SYSRELS TO releaseusr;
GRANT SELECT ON SYSIBM.SYSRELS TO communityusr;
GRANT SELECT ON SYSIBM.SYSRELS TO customizationusr;

GRANT SELECT ON SYSIBM.SYSRELS TO jcr;
GRANT SELECT ON SYSIBM.SYSRELS TO fdbkdbus;
GRANT SELECT ON SYSIBM.SYSRELS TO lmdbusr;
GRANT USE OF STOGROUP jcrstogroup TO jcr;
GRANT CREATEIN, DROPIN ON SCHEMA jcrschema TO jcr;
GRANT SELECT ON SYSIBM.SYSTABLESPACE TO jcr;
GRANT SELECT ON SYSIBM.SYSVIEW TO jcr;
GRANT SELECT ON SYSIBM.SYSDUMMY1 TO jcr;
GRANT SELECT ON SYSIBM.SYSTRIGGERS TO jcr;
GRANT SELECT ON SYSIBM.SYSINDEXPART TO jcr;
GRANT SELECT ON SYSIBM.SYSINDEXES TO jcr;
GRANT SELECT ON SYSIBM.SYSSYNONYMS TO jcr;

where:
2024
- releasenameonzos, communitynameonzos, customizationnameonzos, and releasusr, communityusr, customizationusr represent the databases and database users, respectively, of the WebSphere Portal instance you are setting up. (These users must be created on the host system.)
- jcrdbnameonzos and jcr are the database and database user, respectively, for Content Repository data.
- fdbkdbnameonzos and feedback are the database and database user, respectively, for Feedback data.
- lmdbnameonzos and lmdbusr are the database and database user, respectively, for Likeminds data.
- jcrschema is the database schema name for Content Repository data.

3. Grant the necessary access rights to all users who might require them. Depending on the architecture that you choose, these users might include Java Content Repository and Feedback users.

Parent topic: Configuring WebSphere Portal to use DB2 for z/OS
Previous topic: Installing DB2 for z/OS
Next topic: Creating remote databases
Creating remote databases

A remote database resides on a different machine than WebSphere Portal. You must manually create the required databases before configuring WebSphere Portal to work with DB2 for z/OS.

Before you begin, complete the following prerequisites:

- These instructions assume that you will use a local IBM® DB2 Universal Database™ Enterprise Server Edition Connect to connect to a remote IBM DB2 Universal Database for z/OS® server, and that IBM WebSphere® Application Server, WebSphere Portal, and DB2 Connect will be installed on the same machine.
- To run SQL statements, you can use a tool like SPUFI. The specified statements include CREATE STOGROUP, CREATE DATABASE and CREATE TABLESPACE. For CREATE STOGROUP you have to replace the `icmvolumes` and `icmvcat` variables with volume serial numbers and a catalog name for your environment. For CREATE TABLESPACE you can specify a specific BUFFERPOOL. Refer to the DB2 for z/OS SQL Reference for a description of these options.
- Ensure that a TEMP database is created for your subsystem. You can use the following statements to create a TEMP database:

```
CREATE DATABASE db_name AS TEMP;
CREATE TABLESPACE ts_name IN db_name;
```

- Log on to the DB2 subsystem on the host server. DB2 system administrator rights are needed to create the databases.
- Replace variables as follows:

  - `releasenameonzos`, `communitynameonzos`, and `customizationnameonzos` are the WebSphere Portal databases for WebSphere Portal and Member Manager data.
  - `fdbkdbnameonzos` and `fdbkdbts` are the database and table space, respectively, for Feedback data.
  - `lmdbnameonzos` and `lmdbts` are the database and table space, respectively, for LikeMinds data.
- Because large objects are stored in columns that can become very large, logging changes to these columns requires a huge amount of log space. For this reason, large object (LOB) logging is disabled by default for table spaces that contain such data. With LOB logging disabled, you can recover full backups, but not incremental backups that can be used for point in time recovery. To recover point in time backups, you must enable LOB logging. For detailed instructions, see Technote 1306637, Managing LOB logging in DB2 for z/OS.

**Prerequisites**

- Installing DB2 for z/OS
- Creating users

Use the following steps to set up WebSphere Portal with a remote instance of DB2 for z/OS. Refer to Planning for DB2 for z/OS for a list of databases and table space names. If you are configuring multiple WebSphere Portal instances to use a single DB2 subsystem, be sure to use the database and table space names associated with the WebSphere Portal instance you are setting up.

1. CREATE DATABASE `releasenameonzos` CCSID UNICODE;
2. CREATE DATABASE `communitynameonzos` CCSID UNICODE;
3. CREATE DATABASE `customizationnameonzos` CCSID UNICODE;
4. Execute the steps in the topic Creating the Java Content Repository database.
5. CREATE DATABASE `fdbkdbnameonzos` CCSID UNICODE;
6. CREATE TABLESPACE `fdbkdbts` IN `fdbkdbnameonzos` USING STOGROUP SYSDEFLT PRIQTY 5000 SECQTY 500 LOCKSIZE ROW;
7. CREATE DATABASE `lmdbnameonzos` CCSID UNICODE;
CREATE TABLESPACE lmdbts IN lmdbnameonzos USING STOGROUP SYSDEFLT PRIQTY 5000 SECQTY 500;

Parent topic: Configuring WebSphere Portal to use DB2 for z/OS
Previous topic: Creating users
Next topic: Creating the Java Content Repository database

Related Information
- Managing LOB logging in DB2 for z/OS
Creating the Java Content Repository database

View information on setting up your Java Content Repository database to work with WebSphere Portal.

First, you need to create multiple databases for scalability. The IBM Java Content Repository database grows in size as the IBM® WebSphere® Portal server is used. To support this growth, which includes the dynamic creation of many new tables within the database, the WebSphere Portal server allows you to create multiple databases within a single IBM DB2 Universal Database™ for z/OS® subsystem. A minimum of two databases is recommended, but more can be created if necessary. A single database setup is not recommended; it will quickly become constrained. See the Performance guides on the product documentation Web site for more information.

Each database that is created must begin with a common prefix; there is no convention required for the remainder of the name. For example, to create \textit{xx} number of databases, you may choose to use the following commands:

\begin{verbatim}
CREATE DATABASE JCRDB01
CREATE DATABASE JCRDB02
...  
CREATE DATABASE JCRDBxx
\end{verbatim}

In this case, \textit{JCR} is the common prefix. You also have to select a maximum number of user-defined tables (UDT) that each database will be allowed to hold; the recommended value is 400.

- **Prerequisites**
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases

A sample script of the commands that you can use to create the Java Content Repository database in your DB2 for z/OS installation is shown below. The sample demonstrates the creation of a single database. To follow the recommendation of creating at least two databases, duplicate the commands for each additional database as indicated below. Find a template of these commands in the file \texttt{PortalServer_root/installer/wconfig/config/templates/db/db2_zos/jcr_sample.sql}.

**Notes:**
- It is not necessary to duplicate every tablespace definition in every database.
- In order to run the commands shown in the sample, you must know the database name and the IDs of the MVS volumes where the Java Content Repository database tables will be stored.
- Edit the template file for your installation environment before running the commands shown in the sample:
  - Replace \texttt{jcrdbnameX} with the name of the database. Remember that each database name must begin with a common prefix.
  - Replace \texttt{stogroup} with the name of your storage group.
  - Replace \texttt{icmvolumes} with the IDs of the MVS volumes where the Java Content Repository database tables will be stored. These values are assigned by the database administrator.
  - Replace \texttt{icmvcat} with the name of the virtual catalog.
  - Replace \texttt{jcr} with the name of database user ID.
  - Replace \texttt{4kbp} with the name of your 4K bufferpool.
  - Replace \texttt{32kbp} with the name of your 32K bufferpool.
  - Replace \texttt{jcrschema} with the schema name of your Java Content Repository domain.
CREATE DATABASE jcrdbname

CREATE STOGROUP stogroup VOLUMES (icmvolumes) VCAT icmvcat

GRANT USE OF STOGROUP stogroup to jcr

CREATE DATABASE jcrdbname STOGROUP stogroup BUFFERPOOL 4kbp INDEXBP 4kbp CCSID UNICODE

Note: The following two tablespace commands must be created in every database:

CREATE TABLESPACE ICMLFQ32 IN jcrdbname USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE

CREATE TABLESPACE ICMTSQ04 IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

Note: The following tablespaces are required in the first database only:

CREATE TABLESPACE ICMLFQ32 IN jcrdbname USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE

CREATE TABLESPACE ICMTSQ04 IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ICSTADO IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE PRIDLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE PRSCLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE PRISLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE PRGCLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE PRIGLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ICSTDPV IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ACCCLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE USEGLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ACCLLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE SYSCLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE COLNLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE USERLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE USEGLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ACCLLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE SYSCLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE COLNLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE USERLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 10000 SECQTY 1000 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE USEGLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ACCLLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5 LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE SYSCLSTS IN jcrdbname USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE CACLFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE CPERFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ATTFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ATTGFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE NLSSFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE NLSLFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE NLSKFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ITFLFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ITGFLFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ITFLGFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ITKFLFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ITKFLGFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ITFLFLFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ITGFLFLFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ITFLGFLFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ITKFLFLFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ITKFLGFLFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ITFLFLFLFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ITGFLFLFLFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ITFLGFLFLFLFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ITKFLFLFLFLFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ITKFLGFLFLFLFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

CREATE TABLESPACE ITFLFLFLFLFLFLSTS IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT301 USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE ICMACLS IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE ICMACLC IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE ICMACTS IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00208 IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00207 IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00206 IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00205 IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00204 IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00203 IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00210 IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00202 IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00209 IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00201 IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TS00200 IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TIELLSTS IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE REPLLSTS IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE REPRLSTS IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE RI11LSTS IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE XDOFLSTS IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE XDOOLSTS IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE IDELLSTS IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE TEICLSTS IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
CREATE TABLESPACE SYAELSTS IN USING STOGROUP stogroup PRIOQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

jcrdbnameX

USING STOGROUP

stogroup

CCSID UNICODE

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5

PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT311 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT321 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT331 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT341 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE ICUT601 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT311 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT321 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT331 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT341 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT601 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT311 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT321 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT331 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT341 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT601 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT311 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT321 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT331 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT341 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE TABLESPACE ICUT601 IN jcrdbnameX USING STOGROUP stogroup PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TSPEJRTS IN jcrdbnameX USING STOGROUP stogroup 
   PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE TSTAJRTS IN jcrdbnameX USING STOGROUP stogroup 
   PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE RIDSJRTS IN jcrdbnameX USING STOGROUP stogroup 
   PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE WSNOJRTS IN jcrdbnameX USING STOGROUP stogroup 
   PRIQTY 50000 SECQTY 10000 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 32kbp CCSID UNICODE
CREATE LOB TABLESPACE ICMLOBT1 IN jcrdbnameX LOG NO
CREATE LOB TABLESPACE ICMLOBT2 IN jcrdbnameX LOG NO
CREATE LOB TABLESPACE ICMLOBT3 IN jcrdbnameX LOG NO
CREATE ALIAS jcrschema.ICMFICTITIOUS FOR SYSIBM.SYSDUMMY1
CREATE TABLESPACE DLKRJRTS IN jcrdbnameX USING STOGROUP stogroup 
   PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE
CREATE TABLESPACE LKRLJRTS IN jcrdbnameX USING STOGROUP stogroup 
   PRIQTY 500 SECQTY 200 FREEPAGE 1 PCTFREE 5
LOCKSIZE ROW SEGSIZE 32 LOCKMAX 0 CLOSE NO BUFFERPOOL 4kbp CCSID UNICODE

Parent topic: Configuring WebSphere Portal to use DB2 for z/OS
Previous topic: Creating remote databases
Next topic: Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the wp_profile_root
  /PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.
- When using IBM® DB2 Universal Database™ for z/OS® as a data store, WebSphere Portal requires that its indexes are not padded. Therefore, you must set the DSNZPARM parameters to RETVLCFK=NO or PADIX=NO, or both.

Prerequisites
- Installing DB2 for z/OS
- Creating users
- Creating remote databases

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.index_name.indexspace
   For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds
3. Assign a table space to each .tablespace entry in the mapping file. Assignments to .indexspace entries are ignored.
   The table space name must be qualified by the database name and prepended by the keyword IN and a space. For example: community.COMP_INST.tablespace=IN COMM8KSPACE
   Repeat this step for each domain that you are transferring.
4. Save and close dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
- UNIX: ./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true
- i5/OS: ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true

**Parent topic:** Configuring WebSphere Portal to use DB2 for z/OS  
**Previous topic:** Creating the Java Content Repository database  
**Next topic:** Configuring WebSphere Portal to use DB2 for z/OS
Configuring WebSphere Portal to use DB2 for z/OS

View information on manually transferring data to the DB2 for z/OS you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to DB2 for z/OS. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:
- Supported database software is installed.
- Databases and users are set up.

- Prerequisites
  - Installing DB2 for z/OS
  - Creating users
  - Creating remote databases

Tips:
- If you are transferring from Oracle, the `open_cursors` setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

Steps for transferring data to another supported database

1. If you are running a type 2 connection, before transferring data edit the `db2cli.ini` file. Failure to follow these steps will cause the database transfer to hang at the task `action-process-constraints`.
   A. Locate the file `C:\Program Files\IBM\SQLLIB\db2cli.ini`.
   B. Add the following to the end of the file. Leave an empty line after `ReturnAliases=0`. [COMMON]
      ```
      DYNAMIC=1
      ReturnAliases=0
      ```

2. Open a command prompt and change to the directory `wp_profile_root\ConfigEngine`.

3. Enter the following commands to validate configuration properties.
   ```
   ConfigEngine.bat validate-database-driver -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ConfigEngine.bat validate-database-connection -DTransferDomainList=release,customization,community,jcr,feedback,likeminds -DWasPassword=password
   ```

4. From the same command prompt as the previous steps, change to the directory `wp_profile_root\bin`.

5. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td><code>stopServer.bat server1 -username admin_userid -password admin_password</code></td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td><code>stopServer.bat WebSphere_Portal -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>
6. **Transfer the database:**

   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory `wp_profile_root\ConfigEngine`.

   B. Enter the following command:
   ```cmd
   ConfigEngine.bat database-transfer -DWasPassword=password
   ```

   **Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:
   ```cmd
   ConfigEngine.bat database-transfer -DTransferDomainList=jcr -DWasPassword=password
   ```

   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

    If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

7. From the command line processor, reset the check pending status on the WebSphere Portal table spaces listed here.

   **CHECK DATA** is a DB2 batch utility that is invoked through JCL or through the DB2 utility panels. Type the following utility commands to check pending status:
   ```bash
   check data tablespace releasenameonzos.TS280A
   check data tablespace releasenameonzos.TS300A
   check data tablespace releasenameonzos.TS2110A
   check data tablespace releasenameonzos.TS830A
   check data tablespace communitynameonzos.TS8000B
   check data tablespace communitynameonzos.TS8011B
   check data tablespace communitynameonzos.TS280B
   check data tablespace customizationnameonzos.TS2110C
   check data tablespace jcrdbnameonzos.ICMSFQ04
   check data tablespace jcrdbnameonzos.TS2110D
   ```

   where `releasenameonzos`, `communitynameonzos`, and `customizationnameonzos` are the names of your WebSphere Portal databases, and `jcrdbnameonzos` is the name of your JCR database. Refer to the *Utility Guide and Reference for DB2 for z/OS* for additional details.

8. Run the `RUNSTATS` utility as shown in the following example:
   ```bash
   LISTDEF JCRDBZOS INCLUDE TABLESPACE JCRDBZOS.*  BASE
   RUNSTATS TABLESPACE LIST JCRDBZOS INDEX(ALL) KEYCARD TABLE(ALL)
   LISTDEF JCRDB01 INCLUDE TABLESPACE JCRDB01.*  BASE
   RUNSTATS TABLESPACE LIST JCRDB01 INDEX(ALL) KEYCARD TABLE(ALL)
   ...
   ```

9. Start the WebSphere Application Server.

10. Verify that the WebSphere Portal application server is running by opening the following URL in a browser:

    ```http://hostname.companyname.com:port_number/wps/portal```

    where `hostname.companyname.com` is the fully qualified host name of the machine where WebSphere Portal is running and `port_number` is the transport port that is created by WebSphere Application Server.

If you have performed database transfer in a clustered environment, you must explicitly copy the `wp_profile_root\PortalServer\jcr\lib\com\ibm\icm\icm.properties` file from the primary node on which the `database-transfer` task was processed to all secondary nodes. This ensures that the secondary nodes have the new JCR database values specified in the `icm.properties` file.

1. Stop the portal server on the secondary node.

2. Copy the `wp_profile_root\PortalServer\jcr\lib\com\ibm\icm\icm.properties` file from the primary node and replace the `icm.properties` file on the secondary node with the new file from the primary node.

3. Start the portal server on the secondary node.
Configuring WebSphere Portal to use Oracle

View information on installing and setting up Oracle to work with WebSphere Portal.

1. Installing Oracle
   View information on how to install Oracle for use with WebSphere Portal.

2. Creating users
   View the steps to set up users for Oracle to work with WebSphere Portal.

3. Creating databases
   View some important considerations before setting up Oracle databases to work with WebSphere Portal.

4. Modifying database properties
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. Setting up databases
   This topic provides instructions on automatically setting up your database using the ConfigEngine task to create users, grant permissions, and create Java Content Repository table spaces.

6. Creating JCR table spaces
   This topic provides manual instructions for creating JCR table spaces for Oracle.

7. Optional: Assigning custom table spaces
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

8. Configuring WebSphere Portal to use Oracle
   View information on manually transferring data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Parent topic: Configure WebSphere Portal to use a remote database
Installing Oracle

View information on how to install Oracle for use with WebSphere Portal.

You can use Oracle as the database software. You must install Oracle separately and before performing a database transfer.

Refer to Oracle product documentation for installation instructions.

**Parent topic:** Configuring WebSphere Portal to use Oracle

**Next topic:** Creating users
Creating users

View the steps to set up users for Oracle to work with WebSphere Portal.

Before you begin:

- You should have installed Oracle.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

**Prerequisites**

- Installing Oracle

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Start the Oracle SQL command line processor 'SQL*Plus' by entering `sqlplus /nolog` on the operating system command prompt
2. Log in to the Oracle database by executing command `connect user/password@database name`. For example: `system/manager@wpsdb` will log the administrative user system with a password of manager into the wpsdb database.
3. Create the WebSphere Portal user `dbdomain.DbUser`, where `dbdomain` is replaced by release, community, or customization.
   A. Log in to the database in which you want to create the new users.
   B. Create the users that you need in the appropriate databases by entering the following commands in the SQL*Plus tool, where `user_tablespace` is the default tablespace that is identified by the database administrator to store user objects and `temp_tablespace` is identified to store temporary objects. All names must be in lower case. `SQL> create user releaseusr identified by password` `default tablespace user_tablespace` `temporary tablespace temp_tablespace;` `GRANT UNLIMITED TABLESPACE TO releaseusr;` `SQL> create user communityusr identified by password` `default tablespace user_tablespace temporary tablespace temp_tablespace;` `GRANT UNLIMITED TABLESPACE TO communityusr;` `SQL> create user customizationusr identified by password` `default tablespace user_tablespace` `temporary tablespace temp_tablespace;` `GRANT UNLIMITED TABLESPACE TO customizationusr;` `SQL> create user jcr identified by password` `default tablespace user_tablespace` `temporary tablespace temp_tablespace;` `GRANT UNLIMITED TABLESPACE TO jcr` 4. Connect to the Feedback database:
   A. Enter the following command: `SQL> connect`
B. Enter `user-name: username/password@dbname` where `username` is an existing administrative user in the database. For example, `system/manager@fdbkdb` will log the administrative user system with a password of manager into the `fdbkdb` database.

C. Create the Feedback user:

```
SQL> create user feedback identified by password
    default tablespace user_tablespace
    temporary tablespace temp_tablespace;
```

GRANT UNLIMITED TABLESPACE TO `feedback`

D. Log out of the command line tool using the command `SQL> exit`.

5. Connect to the LikeMinds database:

A. Enter the following command:

```
SQL> connect
```

B. Enter `user-name: username/password@dbname`, where `username` is an existing administrative user in the database.

   For example, `system/manager@lmdb` will log the administrative user system with a password of manager into the `lmdb` database.

C. Create the LikeMinds user:

```
SQL> create user lmdbusr identified by password
    default tablespace user_tablespace
    temporary tablespace temp_tablespace;
```

GRANT UNLIMITED TABLESPACE TO `lmdbusr`

D. Log out of the command line tool using the command `SQL> exit`.

6. Perform the following steps to verify that you can connect to the database with the user ID and password that you just created:

A. Enter the `SQL> connect` command to connect to the content database.

B. Enter `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example, `system/manager@jcrdb` will log the administrative user system with a password of manager into the Oracle database.

---

Parent topic: Configuring WebSphere Portal to use Oracle

Previous topic: Installing Oracle

Next topic: Creating databases
Creating databases

View some important considerations before setting up Oracle databases to work with WebSphere Portal.

For information about creating databases, refer to the Oracle product documentation. For information on the recommended database architecture and the databases you will need to create, see the Planning for Oracle topic. Be sure that all databases to be used with WebSphere Portal are created as UNICODE character set databases.

If you are using remote Oracle 9 or 10 databases, you must also copy the ojdbc14.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM® WebSphere® Application Server Version 7.0, you must also copy the ojdbc6.jar file from the remote Oracle server to the WebSphere Portal machine. If you are using remote Oracle 11 database with IBM WebSphere Application Server Version 6.1, you must also copy the ojdbc5.jar file from the remote Oracle server to the WebSphere Portal machine. The typical location is the oracle_home/sqldeveloper/jdbc/lib directory. Record the copy location on your local machine for future reference.

When creating Oracle databases for use with WebSphere Portal, you should consider the following information:

- The Oracle databases must be created manually before configuring WebSphere Portal.
- All databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle server will be remote or local to the WebSphere Portal installation.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle database in order for WebSphere Portal to communicate with the Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes:

  - `db_block_size` = 8192 bytes
  - `db_cache_size` = 307,200 bytes
  - `db_files` = 1024 files
  - `log_buffer` = 65536 bytes
  - `open_cursors` = 1500 cursors
  - `pga_aggregate_target` = 204,800 bytes
  - `pre_page_sga` = true
  - `processes` = 300 processes
  - `shared_pool_size` = 204,800 bytes

  **Note:** If you are using IBM Java Content Repository, the `open_cursors` value may need to be increased based on the table count in the Java Content Repository schema.

- Raise the number of parallel servers as appropriate. For example, if you have more than 875 parallel servers, you should set the `parallel_max_servers` to 1200.

- The Oracle parameter CURSOR_SHARING allows similar SQL Statements to be shared when possible, which prevents parsing and establishing a new execution plan. The execution plan is used by Oracle to gather the data needed to satisfy a request. There are two options for CURSOR_SHARING, which are as follows:
  - **FORCE**
    - When you select this option, Oracle uses the same execution plan for all SQLs that are similar in value even if the values are different. When you use this option, the execution plan may not provide optimum performance. For example, similar SQLs with different values may behave differently when executed running the same plan.
  - **EXACT**
When you select this option, Oracle only shares the same execution plan for SQLs that are identical and use the same values. This option removes the risk of a SQL statement being executed when optimum performance conditions do not exist.

WebSphere Portal supports both options. Regardless of the option selected, portlet applications should not be affected. Contact your database administrator for further assistance on these options.

**Prerequisites**

- Installing Oracle
- Creating users

**Parent topic:** Configuring WebSphere Portal to use Oracle

**Previous topic:** Creating users

**Next topic:** Modifying database properties
Modifying database properties
Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- Prerequisites
  - Installing Oracle
  - Creating users
  - Creating databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

Tip: For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

Working with properties files:
- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as release.DbName, jcr.DbName, feedback.DbName, and likeminds.DbName. For example:
  - release.DbName=release
  - jcr.DbName=jcrdb
  - feedback.DbName=fdbkdb
  - likeminds.DbName=lmdb
  - community.DbName=commdb
  - customization.DbName=custom
- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.
- Depending on which database domain has to be configured, replace dbdomain with:
  - release
  - customization
  - community
  - jcr
  - feedback
  - likeminds
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
  - dbdomainDbType
- `dbdomain.DbName`
- `dbdomain.DbUrl`
- `dbdomain.DbSchema`

If you use the same values for all four properties across the release, customization, community, and JCR domains, the database-transfer task fails because of ambiguous database object names. If `DbUser`, `DbUrl`, and `DbPassword` are not the same across domains, the value for `DataSourceName` must differ from the `DataSourceName` of the other domains. In other words, this value must be unique for the database domain.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

1. Locate the following files and create a backup copy of each before changing any values:
   - `wp_profile_root/ConfigEngine/properties/wkplc.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_comp.properties`
   - `wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties`

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In `wkplc_comp.properties`, most properties are repeated for each domain.

2. Use a text editor to open the properties file `wkplc_comp.properties` and modify the values to correspond to your environment.
   
   A. For `dbdomainDbType`, type `oracle`.
   
   B. For `dbdomain.DbName`, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the `dbdomain.DbUrl` property.
   
   C. For `dbdomain.DbSchema`, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The `dbdomain.DbName` should be the same value used for the `dbdomain.DbSchema`.

       **Restriction:** The value for `dbdomain(DbSchema)` must equal the value for `dbdomain(DbUser)`.

   D. For `dbdomain.DataSourceName`, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
   - `releaseDS`
   - `communityDS`
   - `customizationDS`
   - `jcrDS`
   - `lmdbDS`
   - `feedback`

   E. For `dbdomain.DbUrl`, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:**
   - The database element of this value should match the value of `DbName`.
   - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following:
     
     ```
     jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME.
     ```
     
     When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.

   F. For `dbdomain.DbUser`, type the user ID for the database administrator. **Restriction:** The value for `dbdomain(DbUser)` must equal the value for `dbdomain(DbSchema)`.
G. For `dbdomain.DbPassword`, type the password for the database administrator.
H. For `dbdomain.DBA.DbUser`, type the database administrator user ID for privileged access operations during creation of the database.
I. For `dbdomain.DBA.DbPassword`, type the database administrator password for privileged access operations during creation of the database.
J. For `dbdomain.DbHome`, type the root location for the database. **Note:** This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in `wkplc_comp.properties`. **Important:** The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.
A. For `source.domain.DbType`, type of the database you are currently configured to use. The value for `source.domain.DbType` is Derby by default.
B. For `source.domain.DbName`, type the name of the database domain you are currently using.
C. For `source.domain.DbSchema`, type current schema identifier for objects within the database for this domain.
D. For `source.domain.DataSourceName`, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.
E. For `source.domain.DbUrl`, type the url currently used to access your database.
F. For `source.domain.DbUser`, type the name of the user accessing this database.
G. For `source.domain.DbPassword`, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file `wkplc_dbtype.properties`.
   A. For `oracle.DbDriver`, type the name of the class that SqlProcessor uses to import SQL files.
   B. For `oracle.DbLibrary`, type the directory and name of the .zip or .jar file that contains the JDBC driver class.
   C. For `oracle.JdbcProviderName`, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file `wkplc.properties`.
   A. For `WasPassword`, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

**Parent topic:** Configuring WebSphere Portal to use Oracle

**Previous topic:** Creating databases

**Next topic:** Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task to create users, grant permissions, and create Java Content Repository table spaces.

- **Prerequisites**
  - Installing Oracle
  - Creating users
  - Creating databases
  - Modifying database properties

As an alternative to automatically setting up the database, you can manually set up your database by referring to the link in the related tasks section of this topic.

1. On the database server, make sure that the folder `db_1` contains the subfolders `your_oracle_instance/data` and `your_oracle_instance/index`. If this folder hierarchy does not exist, create it manually before you run the `setup-database` task. The `setup-database` task requires these folders to create database users. If these folders do not exist, the `setup-database` task will fail.

2. Change to the directory `wp_profile_root/ConfigEngine`

3. To create the database users, type the following command:

   ```
   ConfigEngine.bat setup-database -DWasPassword=password
   ```

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires `db_owner` permission for the database user. The script that creates users is invoked by the configuration task `setup-database` and is located in the installation directory of WebSphere Portal:

   `<WP_root>/base/wp.db.impl/config/templates/setupdb/sqlserver2005\<database domain>\createUsers.sql`.

Parent topic: Configuring WebSphere Portal to use Oracle

Previous topic: Modifying database properties

Next topic: Creating JCR table spaces

Related tasks
Manually creating users and granting privileges for Oracle
Creating JCR table spaces

This topic provides manual instructions for creating JCR table spaces for Oracle.

- Prerequisites
  - Installing Oracle
  - Creating users
  - Creating databases
  - Modifying database properties
  - Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.
2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:
   - `&jcrdb` is the name of the database you created to store user data.
   - `&dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.
   - Ensure that the `.` is included in the variables when you substitute the values of your environment with these variables.

   **Important:** You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

```sql
create tablespace ICMLFQ32 datafile '&dbpath./&jcrdb./data/&jcrdb._ICMLFQ32_01.dbf' size 300M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
create tablespace ICMLNF32 datafile '&dbpath./&jcrdb./data/&jcrdb._ICMLNF32_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
create tablespace ICMVFQ04 datafile '&dbpath./&jcrdb./data/&jcrdb._ICMV04_01.dbf' size 25M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
create tablespace ICMSFQ04 datafile '&dbpath./&jcrdb./data/&jcrdb._ICMSFQ04_01.dbf' size 150M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
create tablespace ICMLNDFX datafile '&dbpath./&jcrdb./index/&jcrdb._ICMLNDFX_01.dbf' size 10M reuse autoextend on next 10M maxsize UNLIMITED extent management local autoallocate;
```

A. Set the `size`, `autoextend`, and `maxsize` values according to your environment. For example, you may want to change the `maxsize` to a set value rather than `UNLIMITED`.
B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.
C. Refer to the Oracle command reference for more information about using the `create tablespace` command.
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:

- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

- Prerequisites
  - Installing Oracle
  - Creating users
  - Creating databases
  - Modifying database properties
  - Setting up databases
  - Creating JCR table spaces

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file wp_profile_root/PortalServer/config/tablespaces/dbdomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - dbdomain.table_name.tablespace
   - dbdomain.table_name.index_name.indexspace
   
   For the file name and each table space and index space property pair, dbdomain can be any one of the following values:
   - release
   - community
   - customization
   - jcr
   - feedback
   - likeminds
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword TABLESPACE and a space. For example: community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE
   
   Repeat this step for each domain that you are transferring.
4. Save and close dbdomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
- **UNIX**: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- **i5/OS**: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

**Parent topic:** Configuring WebSphere Portal to use Oracle  
**Previous topic:** Creating JCR table spaces  
**Next topic:** Configuring WebSphere Portal to use Oracle
Configuring WebSphere Portal to use Oracle

View information on manually transferring data to the Oracle database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:
Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

**Prerequisites**
- Installing Oracle
- Creating users
- Creating databases
- Modifying database properties
- Setting up databases
- Creating JCR table spaces

**Tips:**
- If you are transferring from Oracle, the `open_cursors` setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

**Steps for transferring data to another supported database**

1. Open a command prompt and change to the directory `wp_profile_root\ConfigEngine`.
2. Enter the following commands to validate configuration properties. `ConfigEngine.bat validate-database-driver` -
   `DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password`
3. From the same command prompt as the previous steps, change to the directory `wp_profile_root\bin`.
4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td><code>stopServer.bat server1 -username admin_userid -password admin_password</code></td>
</tr>
</tbody>
</table>
5. Transfer the database. **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory `wp_profile_root\ConfigEngine`.

   B. Enter the following command:

   ```
   ConfigEngine.bat database-transfer -DWasPassword=password
   ```

   **Note:** To select specific database domains to transfer, modify the `-DTransferDomainList` specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:

   ```
   ConfigEngine.bat database-transfer -DTransferDomainList=jcr -DWasPassword=password
   ```

   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

   If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example:

   ```sql
   SQL> execute dbms_stats.gather_schema_stats(ownname=>'jcr', cascade=>TRUE);
   ```

7. Change to the directory `wp_profile_root\bin`.

8. Enter the following command to start the WebSphere Portal server:

   ```
   startServer.bat WebSphere_Portal -username admin_userid -password admin_password
   ```

**Parent topic:** Configuring WebSphere Portal to use Oracle

**Previous topic:** Assigning custom table spaces
Configuring WebSphere Portal to use Oracle RAC

View information on installing and setting up Oracle RAC to work with WebSphere Portal.

1. **Installing Oracle RAC**
   View information on how to install Oracle RAC for use with WebSphere Portal.

2. **Create users**
   View the steps to set up users for Oracle RAC to work with WebSphere Portal.

3. **Creating databases**
   View some important considerations before setting up Oracle RAC databases to work with WebSphere Portal.

4. **Modifying database properties**
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

5. **Setting up databases**
   This topic provides instructions on automatically setting up your database using the ConfigEngine task to create users, grant permissions, and create Java Content Repository table spaces. As an alternative to automatically setting up the database, you can manually set up your database.

6. **Creating JCR table spaces**
   This topic provides manual instructions for creating JCR table spaces for Oracle.

7. **Optional: Assigning custom table spaces**
   The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

8. **Configure WebSphere Portal to use Oracle RAC**
   View information on manually transferring data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

**Parent topic:** Configure WebSphere Portal to use a remote database
Installing Oracle RAC

View information on how to install Oracle RAC for use with WebSphere Portal.

Before you begin:
- You should have completed reviewing the Planning for Oracle RAC section.
- Ensure the Oracle CRS (Cluster Ready Service) and Oracle RAC databases have been installed and configured on the primary and secondary nodes.
- Run the following commands on both RAC nodes to start global services daemon (GSD), oracle listeners, and agents.
  - $ gsdctl start
  - $ lsnrctl start
  - $ agentctl start

You can use Oracle RAC as the database software for WebSphere Portal. You must install Oracle RAC separately and before performing WebSphere Portal database transfer.

Refer to Oracle product documentation for installation instructions.

Parent topic: Configuring WebSphere Portal to use Oracle RAC
Next topic: Create users
Create users

View the steps to set up users for Oracle RAC to work with WebSphere Portal.

Before you begin:

- You should have completed installing Oracle RAC.
- If WebSphere Portal Version 6.1.5 and an earlier version of WebSphere Portal coexist, the database user IDs for WebSphere Portal Version 6.1.5 must be different than earlier versions to avoid conflicts during installation.

- Prerequisites
  - Installing Oracle RAC

Ensure that you create users and grant the appropriate privileges in Oracle before configuring WebSphere Portal to work with Oracle.

Take care to create users in an environment that has the same settings as the actual runtime environment. For example, avoid creating a user in an English environment if you plan to use that user in a Turkish environment.

1. Create the users in the WebSphere Portal databases
   A. Log in to the database in which you want to create the new users.
   B. Create the users that you need in the appropriate databases by entering the following command in the SQL*Plus tool, where user_tablespace is the default tablespace that is identified by the database administrator to store user objects and temp_tablespace is identified to store temporary objects:

   \[
   \text{SQL} > \text{create user username identified by password default tablespace user_tablespace temporary tablespace temp_tablespace;}
   \]

   Tip: Balance the storage of user objects among tablespaces to prevent running out of space with overuse of user_tablespace. Also consider increasing the size of user_tablespace when handling a high volume of users.

2. Log in by entering the command \$\text{sqlplus} in SQL*Plus:

3. Enter the command \texttt{user-name: username/password@dbname}, where \texttt{username} is an existing administrative user in the database. For example: system/manager@wpsdb will log the administrative user system with a password of manager into the wpsdb database.

4. Create the WebSphere Portal user \texttt{dbdomain.DbUser}, where \texttt{dbdomain} is replaced by release, community, customization, or JCR. All names must be in lower case. The following code uses the default user tablespace and the default temporary tablespace. Explicitly specify these tablespaces if you do not want to use the defaults.

   \[
   \text{SQL} > \text{create user releaseusr identified by password;}
   \text{SQL} > \text{create user communityusr identified by password;}
   \text{SQL} > \text{create user customizationusr identified by password;}
   \text{SQL} > \text{create user jcr identified by password;}
   \]

5. Connect to the content database by entering the command \texttt{SQL > connect}.

6. Enter the following, where \texttt{username} is an existing administrative user in the database \texttt{user-name: username/password@dbname}.

   For example, system/manager@jcrdb will log the administrative user system with a password of manager into the Oracle RAC database.

7. Create the Oracle RAC users by entering the following: When creating the \texttt{jcr} user, grant all necessary privileges. If you do not grant privileges, you will receive the error ICMADMIN lacks CREATE SESSION Privilege logon denied.
8. Connect to the Feedback database:
   A. Enter the following command: `SQL> connect`
   B. Enter `user-name: username/password@dbname` where `username` is an existing administrative user in the database.
      For example, `system/manager@fdbkdb` will log the administrative user `system` with a password of `manager` into the `fdbkdb` database.

9. Create the Feedback user:
   `SQL> create user feedback identified by password default tablespace users temporary tablespace temp;`

10. Connect to the LikeMinds database:
    `SQL> connect`

11. Enter `user-name: username/password@dbname`, where `username` is an existing administrative user in the database. For example, `system/manager@lmdb` will log the administrative user `system` with a password of `manager` into the `lmdb` database.

12. Create the LikeMinds user:
    `SQL> create user lmdbusr identified by password default tablespace users temporary tablespace temp;`

13. Log out of the command line tool using the command `SQL> exit`.

Parent topic: Configuring WebSphere Portal to use Oracle RAC
Previous topic: Installing Oracle RAC
Next topic: Creating databases
Creating databases

View some important considerations before setting up Oracle RAC databases to work with WebSphere Portal. When creating Oracle RAC databases for use with WebSphere Portal, you should consider the following information:

- The Oracle RAC databases must be created manually before configuring WebSphere Portal.
- All databases must be created using UNICODE Database and National character sets such as UTF8, AL32UTF8, or AL16UTF16.
- It is recommended that all databases to be used with WebSphere Portal are configured in Dedicated Server Mode.
- Determine if your Oracle RAC server will be remote or local to the WebSphere Portal installation.
- If using an earlier version of Oracle RAC (9i or earlier), ensure that Oracle RAC JVM is also installed.
- After installing the database software for WebSphere Portal, you will need to set the buffer pools allocated to the Oracle RAC database in order for WebSphere Portal to communicate with the IBM Java Content Repository database. Use the following recommended values as a guide. Refer to the Oracle product documentation for information on how to set the buffer pools. Recommended initial buffer pool sizes are:

  - `db_block_size` = 8192 bytes
  - `db_cache_size` = 314,572,800 bytes
  - `db_files` = 1024 files
  - `log_buffer` = 65536 bytes
  - `open_cursors` = 1500 cursors
  - `pga_aggregate_target` = 209,715,200 bytes
  - `pre_page_sga` = true
  - `processes` = 300 processes
  - `shared_pool_size` = 209,715,200 bytes

  **Note:** If you are using IBM Java Content Repository, the `open_cursors` value may need to be increased based on the table count in the IBM Java Content Repository schema.

**Prerequisites**

- Installing Oracle RAC
- Create users
Modifying database properties

Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

- **Prerequisites**
  - Installing Oracle RAC
  - Create users
  - Creating databases

The wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties are published in the IBM® WebSphere® Portal wiki. The wiki articles include examples, valid values, and descriptions to help you determine what value to provide for a given property. The documentation team and other IBM teams will update the properties to provide more examples and clarity as needed.

**Tip:** For the most up-to-date information about properties, such as examples, valid values, and more, see the following wiki articles:
- WebSphere Portal 6.1.5 wkplc.properties file reference
- WebSphere Portal 6.1.5 wkplc_comp.properties file reference
- WebSphere Portal 6.1.5 wkplc_dbtype.properties file reference

**Working with properties files:**

- The WebSphere Portal database can be used to hold information for applications such as Feedback and LikeMinds. Use similar naming conventions for property values such as `release.DbName`, `jcr.DbName`, `feedback.DbName`, and `likeminds.DbName`. For example:

  - `release.DbName=release`
  - `jcr.DbName=jcrdb`
  - `feedback.DbName=fdbkdb`
  - `likeminds.DbName=lmdb`
  - `community.DbName=commdb`
  - `customization.DbName=custdb`

- If you are using a remote database, enter the values for the remote server.
- Use a forward slash (/) instead of a backslash (\).
- There might be additional database properties other than those listed here. Only change the properties within this task and skip all other properties.
- The recommended value listed for each property represents the specific information that is required to configure WebSphere Portal to your target database.

- Depending on which database domain has to be configured, replace `dbdomain` with:
  - `release`
  - `customization`
  - `community`
  - `jcr`
  - `feedback`
  - `likeminds`
- The values for at least one of the following properties must be unique for the release, customization, community, and jcr domains:
  - `dbdomain DbType`
1. Locate the following files and create a backup copy of each before changing any values:
   - wp_profile_root/ConfigEngine/properties/wkplc.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_comp.properties
   - wp_profile_root/ConfigEngine/properties/wkplc_dbtype.properties

   Default values are listed in these files. Unless otherwise noted, all values are of type alphanumeric text string. Print out the steps below for reference before modifying the properties files. Make sure to enter the appropriate values for each instance of each property. In wkplc_comp.properties, most properties are repeated for each domain.

2. Use a text editor to open the properties file wkplc_comp.properties and modify the values to correspond to your environment.

   A. For dbdomainDbType, type oracle.

   B. For dbdomainDbName, type the name of the WebSphere Portal domain database. **Note:** This value is also the database element in the dbdomainDbUrl property.

   C. For dbdomainDbSchema, type the schema name of the database domain. **Note:** Review your target database management system documentation to define a valid schema name. Some database management systems have schema name restrictions that you need to understand. The dbdomainDbName should be the same value used for the dbdomainDbSchema.

     **Restriction:** The value for dbdomainDbSchema must equal the value for dbdomainDbUser.

   D. For dbdomainDataSourceName, type the name of the data source that WebSphere Portal uses to communicate with its databases. Do not use the following reserved words:
      - releaseDS
      - communityDS
      - customizationDS
      - jcrDS
      - lmdbDS
      - feedback

   E. For dbdomainDbUrl, type the database URL used to access the WebSphere Portal database with JDBC. The value must conform to the JDBC URL syntax specified by the database. **Note:**
      - The database element of this value should match the value of DbName.
      - For Oracle RAC only, the WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following:
        jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME. When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.

   F. For dbdomainDbUser, type the user ID for the database administrator. **Restriction:** The value for dbdomainDbUser must equal the value for dbdomainDbSchema.
G. For dbdomain.DbPassword, type the password for the database administrator.
H. For dbdomain.DBA.DbUser, type the database administrator user ID for privileged access operations during creation of the database.
I. For dbdomain.DBA.DbPassword, type the database administrator password for privileged access operations during creation of the database.
J. For dbdomain.DbHome, type the root location for the database. Note: This value is used to specify the location to create the tablespaces.

3. (Optional) When transferring WebSphere Portal databases from any database other than the default Derby database, update the following properties in wkplc_comp.properties. Important: The default values for the following parameters are set for the out-of-the-box Derby configuration. If you are configuring WebSphere Portal for an external database for the first time, do not modify these properties, and skip this step. If you have already configured WebSphere Portal to use a database other than the default Derby database, update the source properties for your current database settings.
A. For source.domainDbType, type of the database you are currently configured to use. The value for source.domainDbType is Derby by default.
B. For source.domainDbName, type the name of the database domain you are currently using.
C. For source.domainDbSchema, type current schema identifier for objects within the database for this domain.
D. For source.domainDataSourceName, type the name of the datasource that is currently used in your IBM WebSphere Application Server configuration.
E. For source.domainDbUrl, type the url currently used to access your database.
F. For source.domainDbUser, type the name of the user accessing this database.
G. For source.domainDbPassword, type the password of the source DbUser.

4. Save and close the file.

5. Update the following properties in the file wkplc_dbtype.properties.
   A. For oraclegetDbDriver, type the name of the class that SqlProcessor uses to import SQL files.
   B. For oraclegetDbLibrary, type the directory and name of the .zip or .jar file that contains the JDBC driver class.
   C. For oracleJdbcProviderName, type the name of the JDBC provider that WebSphere Portal uses to communicate with its databases.

6. Save and close the file.

7. Update the following property in the file wkplc.properties.
   A. For WasPassword, type the password for the WebSphere Application Server security authentication used in your environment.

8. Save and close the file.

Parent topic: Configuring WebSphere Portal to use Oracle RAC
Previous topic: Creating databases
Next topic: Setting up databases
Setting up databases

This topic provides instructions on automatically setting up your database using the ConfigEngine task to create users, grant permissions, and create Java Content Repository table spaces. As an alternative to automatically setting up the database, you can manually set up your database.

Before you begin, ensure that the following prerequisites are met:

- Supported database software is installed.
- Property files are modified.
- Database user names and passwords should not contain spaces and should comply with both the database management system software requirements and WebSphere Portal requirements for administering users and groups.

**Prerequisites**

- Installing Oracle RAC
- Create users
- Creating databases
- Modifying database properties

1. On the database server, make sure that the folder db_1 contains the subfolders your_oracle_instance/data and your_oracle_instance/index. If this folder hierarchy does not exist, create it manually before you run the setup-database task. The setup-database task requires these folders to create database users. If these folders do not exist, the setup-database task will fail.

2. Change to the directory wp_profile_root/ConfigEngine

3. To create the database users, type the following command:

   `ConfigEngine.bat setup-database -DWasPassword=password`

   **Note:** This task will automatically create the necessary users, permissions, and table spaces. WebSphere Portal requires db_owner permission for the database user. The script that creates users is invoked by the configuration task setup-database and is located in the installation directory of WebSphere Portal: `\<WP_root>\base\wp.db.impl\config\templates\setupdb\sqlserver2005\<database domain>\createUsers.sql`.

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC

**Previous topic:** Modifying database properties

**Next topic:** Creating JCR table spaces

**Related tasks**

Manually creating users and granting privileges for Oracle
Creating JCR table spaces

This topic provides manual instructions for creating JCR table spaces for Oracle.

- **Prerequisites**
  - Installing Oracle RAC
  - Create users
  - Creating databases
  - Modifying database properties
  - Setting up databases

Refer to the following instructions for creating tablespaces:

1. In the database directory, create the data directory `data` and the index directory `index`.

2. Create tablespaces using the following commands as examples: Substitute the values of your environment for the following variables:
   - `&jcrdb` is the name of the database you created to store user data.
   - `&dbpath` is the directory where you created the database; the default path is `/oracle/oradata`.

   Ensure that the `.` is included in the variables when you substitute the values of your environment with these variables.

   **Important:** You must use the same table space names listed in the commands. The table space names cannot be customized or modified.

   ```sql
   create tablespace ICMLFQ32 datafile '&dbpath./&jcrdb./data/&jcrdb._ICMLFQ32_01.dbf' size 300M reuse autoextend on
   next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLNF32 datafile '&dbpath./&jcrdb./data/&jcrdb._ICMLNF32_01.dbf' size 25M reuse autoextend on
   next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMVFQ04 datafile '&dbpath./&jcrdb./data/&jcrdb._ICMVFQ04_01.dbf' size 25M reuse autoextend on
   next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMSFQ04 datafile '&dbpath./&jcrdb./data/&jcrdb._ICMSFQ04_01.dbf' size 150M reuse autoextend on
   next 10M maxsize UNLIMITED extent management local autoallocate;
   create tablespace ICMLSNDX datafile '&dbpath./&jcrdb./index/&jcrdb._ICMLSNDX_01.dbf' size 10M reuse autoextend on
   next 10M maxsize UNLIMITED extent management local autoallocate;
   ``

A. Set the `size`, `autoextend`, and `maxsize` values according to your environment. For example, you may want to change the `maxsize` to a set value rather than `UNLIMITED`.

B. Consult your Database Administrator for specific guidance about creating tablespaces for your environment.

C. Refer to the Oracle command reference for more information about using the `create tablespace` command.

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC  
**Previous topic:** Setting up databases  
**Next topic:** Assigning custom table spaces
Assigning custom table spaces

The repository of WebSphere Portal consists of many tables and indices that are created in default table spaces. When using an existing set of table spaces for the objects of the repository, specify this when executing the database transfer to the target database system.

Before you begin:
- The custom table spaces must exist prior to the execution of database transfer.
- To see which table spaces can be customized in each domain, refer to the \wp_profile_root\PortalServer\config\tablespaces\ddomain.space_mapping.properties file.
- For details on creating table spaces refer to the documentation for the database.

**Prerequisites**
- Installing Oracle RAC
- Create users
- Creating databases
- Modifying database properties
- Setting up databases
- Creating JCR table spaces

If custom table spaces are assigned, each must be assigned explicitly. The default table spaces can be used to contain database objects; however the name of the default table space must be specified in the corresponding mapping files. This applies to all database domains that are transferred in a single database transfer.

To configure custom table space assignments:

1. Determine the names of your custom table spaces.
2. Open the mapping file \wp_profile_root\PortalServer\config\tablespaces\ddomain.space_mapping.properties that specifies the table space and index space property pairs for each database table:
   - \ddomain.table_name.tablespace
   - \ddomain.table_name.index_name.indexspace
   For the file name and each table space and index space property pair, \ddomain\ can be any one of the following values:
     - release
     - community
     - customization
     - jcr
     - feedback
     - likeminds
3. Assign a table space to each entry in the mapping file. The table space name must be prepended by the keyword TABLESPACE and a space. For example: \community.COMP_INST.tablespace=TABLESPACE COMM8KSPACE
   Repeat this step for each domain that you are transferring.
4. Save and close \ddomain.space_mapping.properties.
5. From a command prompt, specify the option -DuseCustomTablespaceMapping=true when starting the database transfer. For example,
   - Windows: ConfigEngine.bat database-transfer -DuseCustomTablespaceMapping=true
- UNIX: `./ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`
- i5/OS: `ConfigEngine.sh database-transfer -DuseCustomTablespaceMapping=true`

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC

**Previous topic:** Creating JCR table spaces

**Next topic:** Configure WebSphere Portal to use Oracle RAC
Configure WebSphere Portal to use Oracle RAC

View information on manually transferring data to the Oracle RAC database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to Oracle. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

Before you begin:

Ensure that the following prerequisites are met:

- Supported database software is installed.
- Databases and users are set up.
- Property files are modified.

**Prerequisites**

- Installing Oracle RAC
- Create users
- Creating databases
- Modifying database properties
- Setting up databases
- Creating JCR table spaces

**Tips:**

- If you are transferring from Oracle, the **open_cursors** setting should be set to 1500 by default. If you are using Java Content Repository, this value might need to be increased based on the table count in the Java Content Repository schema.

- The WebSphere Portal server must explicitly connect to one RAC node during database transfer. You need to specify the information of one Oracle RAC node as if it is the only database server. For example, the Oracle database URL should look like the following:

```
jdbc:oracle:thin:@PRIMARY_NODE_HOSTNAME:1521:PRIMARY_NODE_INSTANCENAME
```

When database transfer is completed, the WebSphere Portal server will be configured to use this single database server.

- Manually specify an RAC datasource URL in the WebSphere Application Server console. The URL should look like the following:

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT=1521)) (ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICE_NAME))
```

- When doing a single database, single user, and multi schema database transfer, there can be only one user for each domain (release, community, customization, JCR, Feedback, and LikeMinds), and the schema for each database must be different. The user must be a superuser or DBA and must have authority over all other schemas for the transfer to work.

**Steps for transferring data to another supported database**

1. Open a command prompt and change to the directory `wp_profile_root\ConfigEngine`.
2. Enter the following commands to validate configuration properties.

   ConfigEngine.bat validate-database-driver -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password

   ConfigEngine.bat validate-database-connection -DTransferDomainList=release,customization,community, jcr,feedback,likeminds -DWasPassword=password

3. From the same command prompt as the previous steps, change to the directory `wp_profile_root\bin`.

4. Stop both WebSphere Application Server and the WebSphere Portal server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td>stopServer.bat server1 -username admin_userid -password admin_password</td>
</tr>
<tr>
<td>WebSphere Portal</td>
<td>stopServer.bat WebSphere_Portal -username admin_userid -password admin_password</td>
</tr>
</tbody>
</table>

5. Transfer the database:

   **Note:** Important: Do not execute the database-transfer task as a background process. This might cause the task to stall.

   A. Change to the directory `wp_profile_root\ConfigEngine`.

   B. Enter the following command:

   `ConfigEngine.bat database-transfer -DWasPassword=password`

   **Note:** To select specific database domains to transfer, modify the -DTransferDomainList specified in the command to include only the domains that you want to transfer. For example, to transfer only the JCR domain you can enter the following command:

   `ConfigEngine.bat database-transfer -DTransferDomainList=jcr -DWasPassword=password`

   C. After running this task, a message is added to the log files to verify that this task was successful. Check the log files.

   If the configuration fails, verify the values in the `wkplc.properties`, `wkplc_comp.properties`, and `wkplc_dbtype.properties` files and then repeat this step.

6. After transferring the database tables, run the `dbms_stats.gather_schema_stats` command to avoid slow database response. Example:

   SQL> execute dbms_stats.gather_schema_stats(ownname=> 'jcr', cascade=> TRUE);

7. Specify the JDBC URL to connect to the cluster:

   A. Login to the WebSphere Application Server Administrator Console

   B. Navigate to Resources > JDBC Providers

   C. If there is a value in the Node field, remove it and click Apply.

   D. For each Oracle JDBC provider, repeat the following steps:

   1. Click the provider name.

   2. Select Data Sources.

   3. Click the name of the data source.

   4. Under the Oracle data source properties, modify the URL attribute according to Oracle RAC configuration:

   ```
   jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICENAME)))
   ```

   5. Save your changes

8. Change to the directory `wp_profile_root\bin`.

9. Enter the following command to start the WebSphere Portal server:

   `startServer.bat WebSphere_Portal`

---

**Parent topic:** Configuring WebSphere Portal to use Oracle RAC

**Previous topic:** Assigning custom table spaces
Configuring WebSphere Portal to use SQL Server

View information on installing and setting up SQL Server to work with WebSphere Portal.

1. **Installing SQL Server**
   View the steps to install SQL Server for use with WebSphere Portal.

2. **Modifying database properties**
   Learn how to modify the wkplc.properties, wkplc_comp.properties, and wkplc_dbtype.properties files to work with your database. Modify these property files before running tasks to create databases, create users, or transfer data.

3. **Setting up databases**
   This topic provides instructions on automatically setting up your database using the ConfigEngine task. As an alternative to automatically setting up the database, you can manually create users and grant privileges.

4. **Optional: Assigning custom filegroups**
   The repository of WebSphere Portal consists of many tables and indices that are created in default filegroups. When using an existing set of filegroups for the objects of the repository, specify this when executing the database transfer to the target management database system.

5. **Configuring WebSphere Portal to use SQL Server 2005**
   View information on manually transferring data to the SQL Server 2005 database you have installed and set up. Follow these steps to transfer WebSphere Portal, and Java Content Repository databases to SQL Server 2005. As an alternative to the manual database transfer procedure described here, you can use the configuration wizard to complete the database transfer task.

**Parent topic:** Configure WebSphere Portal to use a remote database