

Clustering IBM® Mobile Portal Accelerator (MPA) 6.1

Kiran Kumar K.B., MPA testing,
Staff Software Engineer
IBM India Software Labs, Bangalore
November, 2009

This article is titled as *Clustering IBM® Mobile Portal Accelerator* ® (hereafter referred to as MPA in further document). It demonstrates the comprehensive step-by-step description to build Vertical and Horizontal cluster for the MPA 6.1 on AIX with DB2 as the database on an **existing Portal cluster**. It's targeted for testers and customers who are interested in building scalable/High availability solutions on MPA. It is presumed that the reader of the article has a basic understanding of the IBM® WebSphere Portal ® and WebSphere Portal Clustering.

Introduction

IBM Mobile Portal Accelerator (formerly IBM WebSphere Everyplace Mobile Portal Enable) software is increasingly becoming the key component in many industries like telecommunications, banking, finance, insurance and retail. It extends the capabilities of the IBM WebSphere Portal product to an almost limitless range of mobile devices. It provides Web contents, services and applications to mobile devices while maintaining the benefits and advantages of a portal Web site. These benefits include:

- On-screen application integration
- Single sign-on
- Separation of logic, layout, and branding
- Administration using a powerful and easy to use graphical user interface

Clustering technique enables you to create multiple clones of the application server running identical instance of enterprise applications. These enterprise applications can serve the client request concurrently by running on same (Vertical clustering) or different physical machine's (Horizontal clustering) to provide following benefits.

- **Workload Management.** The client requests can be routed automatically to different cluster members (clones) in the cluster. Hence managing the workload efficiently using clustering.
- **Scalability and Performance.** Additional cluster members (clones) can be added/removed from the cluster based on the number incoming client requests. Hence help in managing the resources timely manner.
- **Reliability and Failover.** Cluster members (clones) can be used for planned / unplanned downtime of the application. In case of failure of any cluster member, still the client request can be routed to other active clones and thus minimizing the risk of failovers.

Clustering can be classified into two major categories based on number of clones/cluster members created on the same or different physical machine. The concept of creating multiple clones/members on same machine is called vertical clustering. Though vertical clustering helps to efficiently utilize the machines processing power but it provides only certain level of failures. Creating multiple clones/members on different physical machines is called horizontal clustering and it provides highest level of scalability and failover.

Pre-requisites

MPA is an extension of WebSphere Portal and it is assumed that, the reader already has a working portal cluster with following details

Java Runtime Environment 1.5.0 service release 8 or later.

IBM® WebSphere Portal Server®: 6.1.0.1 (Here after referred as simply WebSphere Portal Server)

IBM ®WebSphere Application Server®: 6.1.0.19 (Here after referred as WAS)

Database: DB2® Enterprise Edition 8.1 Fix Pack 16 or version 9.1 with fix pack 2 or 4 or DB2 Universal Database Workgroup Server Edition version 9.5 with fix pack 1 (Here after referred as DB2)

Note : Check the infocentre for the latest information of supported versions

(http://publib.boulder.ibm.com/infocenter/mpadoc/v6r1m0/topic/com.ibm.websphere.wemp.doc/planning/softwarerequirements_61.html)

Variable names used in the document

This section lists the variable names and their associated directory paths

was_root - /usr/IBM/WebSphere/AppServer

wp_profile_root - /usr/IBM/WebSphere/wp_profile

wp_root - /usr/IBM/WebSphere/PortalServer

db_root - Value specified for the home directory selected during DB2 installation.

db2instance - DB2® instance user.

Installing MPA on Clustered Environment

This section of the article covers setting up the environment for MPA cluster by using already existing WebSphere Portal cluster as mentioned above in pre-requisites. It covers Preparing WebSphere Portal for MCS, Preparing DB2 for Multi Channel Server (MCS), Installing MCS application and finally installing the Mobile WebSphere Portal (MWP). For the purpose of this article, create a Portal cluster having two nodes (say primary and secondary).

Preparing the WebSphere Portal for MCS

Before you begin installing MPA cluster on existing WebSphere Portal Server cluster, make sure its up and running with no errors/issues.

Refer to the WebSphere Portal 6.1.0.1 Information Center topic on clustering.

http://publib.boulder.ibm.com/infocenter/wpdoc/v6r1/index.jsp?topic=/com.ibm.wp.ent.doc_v6101/install/production.html

Now to start up with building MPA cluster environment you need to first copy the file `wp.base.jar` from the primary node's `<wp_root>/base/wp.base/shared/app/` directory to the deployment manager's `<was_root>/lib` directory and then create the MPA install image as discussed below.

In particular, note that fix pack PK81376 for MPA is required for WebSphere Portal if WebSphere Portal version 6.1.0.1 is used.

In order to ensure you have WebSphere Portal running with no errors/exceptions, open the WebSphere Portal Admin page using the browser and login as the Portal admin user and view pages and look into following logs to confirm same.

- `<wp_profile_root>/logs/WebSphere_Portal/SystemOut.log`
- `<wp_profile_root>/logs/WebSphere_Portal/SystemErr.log`

Creating the MPA install image

Unpack the file `wp.mobile.zip` which is used to create Multi-Channel Server (MCS) and Mobile Portal Accelerator install images on your WebSphere Portal server. Use an archive utility to unzip `wp.mobile.zip` to the `/tmp` directory. Install MCS and Mobile Portal Accelerator as a ConfigEngine assembly using the following command:

```
./ConfigEngine.sh -install -assemblyIudd sdd.xml -assemblyRoot /tmp/wp.mobile -installLocation /IBM/WebSphere -offeringName PortalServer
```

Preparing the DB2 database for MCS

MCS requires a separate and distinct database other than the existing databases for WebSphere Portal. The database can reside on the same machine where the cluster node is created or it can reside on a remote machine. Perform the following steps to create and configure the db

1. Log in to database machine as the DB2 instance user. If you are logged in as root, run the following command to change to the DB2 instance user:
su - db2instance_userID
2. Run the `db2start` command to ensure that the DB2 server is started.
3. Create a separate database for MCS, For example “**mcs**”.
db2 create db mcs using codeset UTF-8 territory us

4. Update the mcs database configuration using the following command to increase the db2 transaction logs.
*db2 update db cfg for **mcs** using logfilsiz 10000 logprimary 10 logsecond 10*
5. Connect to the mcs database using the following command.
*db2 connect to **mcs** user db2instance_userID using db2instance_password*
6. If your database server is on the same server as WebSphere Portal, change the working directory to *<wp_root>/repository/db2* Otherwise, copy the file *<wp_root>/wp.mobile/wp.mobile.mcs/repository/db2/create_vm_tables.sql* from WebSphere Portal server to a directory on database server. Then, make that directory as your working directory.
7. Run the following command to create the database tables:
db2 -tf create_vm_tables.sql
8. Run the following command to start the JDBC applet ("net" driver) on port 6789.
db2jstrt 6789

Note: Use db listening port 6789 unless its already in use. This step is not required for DB2 Universal Database Enterprise Server Edition version 9.1 or DB2 Universal Database Workgroup Server Edition version 9.5.

Installing MCS

Perform the following steps to install MCS on WebSphere Portal server.

1. Ensure that the database server and the database listener are running.
2. Add DB2 JDBC driver to the classpath of the WebSphere Portal server using below steps:
 - a. Log in to the WebSphere Application Server Administrative Console by using the URL <http://dmmanagerip:9060/admin>
 - b. Click *Servers > Application Servers*.
 - c. Click the *WebSphere_Portal* server link.
 - d. Click *Server Infrastructure > Process Definition*.
 - e. Click the *Java Virtual Machine* link.
 - f. In the *Classpath* field, add following entries
Example entries follow.
<db_root>/<db2instance>/sqllib/java/db2java.zip
<db_root>/<db2instance>/sqllib/java/db2jcc.jar
<db_root>/<db2instance>/sqllib/java/db2jcc_license_cu.jar
 - g. Click "Apply" and Save the configuration changes.

Follow the same step for all the WebSphere Portal servers in the portal cluster.

3. Run the below command to ensure that WebSphere Application server environment variables are set from *<wp_profile_root>\bin*
. setupCmdLine.sh
Note: There is a space between the period (.) and the rest of the command.

4. Change your working directory to `<wp_profile_root>/ConfigEngine` and run the following command.

```
./ConfigEngine.sh init mcs-deploy-mcs -DMcsRepositoryType=odbc -  
DMcsDbVendor=db_vendor -DMcsDbHost=hostname -  
DMcsDbPort=port_number -DMcsDbSource=db_name -  
DMcsDbUser=db2instance -DMcsDbPassword=db2instance_password -  
DMcsProject=mobile-portal
```

- *db_vendor*: “db2” for DB2 Universal Database Enterprise Server Edition version 8.1 or “db2-type4” for DB2 Universal Database Enterprise Server Edition version 9.1 and DB2 Universal Database Workgroup Server Edition version 9.5
- *hostname*: the fully qualified host name of the database server.
- *port_number*: the port number used by the JDBC applet (“net”) driver on the database server for the Multi-Channel Server database. Use port 6789 unless another application (including another database instance) is already using that port.
- *db_name*: the name of the Multi-Channel Server database
- *db2instance*: the user ID of the DB2 instance user.
- *db2instance_password*: the DB2 instance password for the MCS database
- *mobile-portal* must be the project name for installing MCS.

Note: Make sure the CLASSPATH environment variables contain the DB2 specific jars i.e.

```
<db_root>/<db2instance>/sqllib/java/db2java.zip (not required for DB2  
Universal Database Enterprise Server Edition version 9.1 or DB2 Universal Database  
Workgroup Server Edition version 9.5)
```

```
<db_root>/<db2instance>/sqllib/java/db2jcc.jar
```

```
<db_root>/<db2instance>/sqllib/java/db2jcc_license_cu.jar (not required  
for DB2 Universal Database Enterprise Server Edition version 8.1)
```

5. Using a text editor, open the *ImportPolicies.sh* script file in the `<wp_root>/repository/bin` directory. Append the JDBC path to the end of CLASSPATH entry

```
db_root>/<db2instance>/sqllib/java/db2java.zip:<db_root>/<db2instance>/sql  
lib/java/db2jcc_license_cu.jar:<db_root>/<db2instance>/sqllib/java/db2jcc.jar
```

6. To import default device policies, run the following command:

```
./ConfigEngine.sh init mcs-import-devices -DMcsProject=mobile-portal
```

Note: Its must to specify “mobile-portal” as the project name as it is the root project for MPA.

7. In case of Horizontal cluster, run the below command on all secondary nodes after running the *setupCmdLine.sh* and setting the CLASSPATH.

```
./ConfigEngine.sh init mcs-deploy-mcs -DMcsRepositoryType=odbc -  
DMcsDbVendor=db_vendor -DMcsDbHost=hostname -DMcsDbPort=port_number
```

*-DMcsDbSource=db_name -DMcsDbUser=db2instance -
DMcsDbPassword=db2instance_password -DMcsProject=mobile-portal*

Note: Use the same parameter values which are used to in the step 4 of Installing MCS section.

8. After the last task completes, edit the *ImportPolicies.sh* file for secondary node as given in above step 5.

Note: Don't run the *mcs-import-device* on secondary node as the MCS data is already imported to the database in step 6.

9. Login to Deployment Manager Admin Console and click **Full Resynchronize** for both cluster members/nodes.

Installing MWP

1. If you haven't already done so with a prior step, please run the following command to ensure that WebSphere Application server environment variables are set from `<wp_profile_root>`
`. setupCmdLine.sh`

2. On Primary node, run the following commands from to `<wp_profile_root>/ConfigEngine` to install the Everyplace Mobile Portal extensions
`./ConfigEngine.sh mwp-config-cluster-primary-node`

Note: Make sure the CLASSPATH environment variables contain the DB2 specific jars i.e.

`<db_root>/<db2instance>/sqllib/java/db2java.zip` (not required for DB2 Universal Database Enterprise Server Edition version 9.1 or DB2 Universal Database Workgroup Server Edition version 9.5)

`<db_root>/<db2instance>/sqllib/java/db2jcc.jar`

`<db_root>/<db2instance>/sqllib/java/db2jcc_license_cu.jar` (not required for DB2 Universal Database Enterprise Server Edition version 8.1)

3. Run this task command from the **Primary node**, to import XDIME aggregator policies and generic portlet policies.

`./ConfigEngine.sh mwp-import-policies`

4. Run this task command from the **Primary node** to import Mobile portlet policies.

`./ConfigEngine.sh mwp-import-portlet-policies`

5. In case of Horizontal cluster, run this command on all secondary cluster nodes after running the `<wp_profile_root>\bin\setupCmdLine.sh` and setting the CLASSPATH
`./ConfigEngine.sh mwp-config-cluster-secondary-node`

Note: Importing aggregator policies and portlet policies (step 3&4 of Installing MWP section) is not required for secondary nodes.

6. After Everyplace Mobile Portal is installed on **both** cluster members of the WebSphere Portal cluster, login to the Administrative Console of Deployment Manager and click full resynchronize on all cluster member/node and restart the portal cluster.
7. Configure the Everyplace Mobile Portal extensions for cluster using the below command on primary node.
`./ConfigEngine.sh mwp-config-cluster-all`
8. To activate the mobile portlets, run the below command on primary node.
`./ConfigEngine.sh activate-portlets`

Note: Finally, click on reinitialize button for the Web server plug-in from the Deployment manager's admin console.

Verifying MPA cluster

Login to mobile portal using a browser and if the user succeeds in viewing Manage mobile page as shown in Figure1,the MPA installation has gone fine.

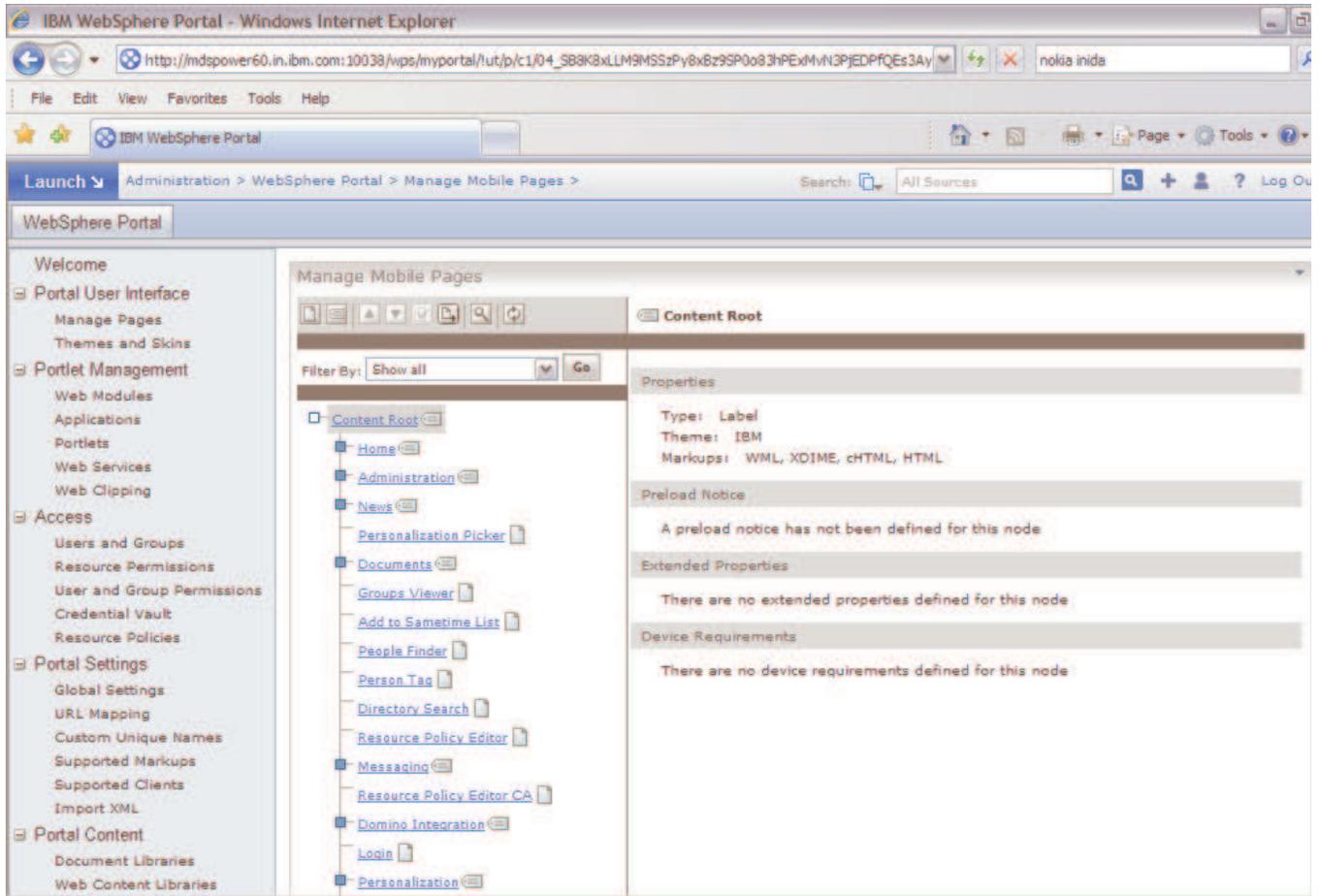


Figure 1. Mobile portal administration page in portal

One can access the same portal url from any mobile, the mobile portal login page should look like below Figure 2:



Figure 2. Mobile portal login page when portal accessed through mobile or mobile simulator

Conclusion

This article demonstrates how to build a MPA cluster using Multi Channel Server and Mobile WebSphere Portal components to build High availability and Scalability solution.

Resources

- More about [Mobile Portal Accelerator 6.1](http://publib.boulder.ibm.com/infocenter/mpadoc/v6r1m0/index.jsp) product and purposes
(<http://publib.boulder.ibm.com/infocenter/mpadoc/v6r1m0/index.jsp>)
- Visit the [Multi-Channel Server Information Center](http://publib.boulder.ibm.com/infocenter/mpadoc/v6r1m0/topic/com.ibm.websphere.wemp.doc/introduction/intro_mcs.html) to learn more about MCS.
(http://publib.boulder.ibm.com/infocenter/mpadoc/v6r1m0/topic/com.ibm.websphere.wemp.doc/introduction/intro_mcs.html)
- Building [WebSphere Portal Cluster](http://publib.boulder.ibm.com/infocenter/wpdoc/v6r1m0/topic/com.ibm.wp.ent.doc_v6101/install/production.html)
(http://publib.boulder.ibm.com/infocenter/wpdoc/v6r1m0/topic/com.ibm.wp.ent.doc_v6101/install/production.html)

About the author

Kiran Kumar K.B. works with WebSphere Everyplace Mobile Portal team as a Staff software engineer. He has acquired extensive knowledge over a period of Six years on WebSphere Portal and MPA. He is currently working on Mobile Portal Accelerator (MPA). You can reach Kiran at kiransagar@in.ibm.com