webtrends™

IBM. **WebSphere.** software

IBM WebSphere Portal Version 6.1 Tracking and Reporting Using Webtrends

Table of Contents

The Webtrends IBM Partnership

In February 2009, Webtrends, a leading provider of web analytics and online marketing solutions, joined the IBM User Experience Optimization Initiative (UEOI).

The initiative addresses the growing demand for better integration between portals and site analyzer tools. The Webtrends-enabled portals will eliminate the need for manually embedding tags into portlets and themes. This makes it easier to measure user activity and make the targeting and design improvements that increase customer satisfaction and decrease infrastructure, tuning and testing costs.

This initiative is the first step toward the creation of smarter, more personalized portal experiences. Webtrends' data collection technologies enable IBM's portal software to accurately pass relevant data from portlets into the SmartSource Data Collection Server. This integration lets you use performance data from Webtrends to optimize WebSphere Portal, and make better marketing decisions using social-based technologies.

The first phase of this initiative will highlight how easy it is to integrate WebSphere Portal and Webtrends technology. This document provides an introduction to Webtrends and IBM WebSphere Portal technology, along with step-by-step guidance on how to collect the proper data and report on it. You will also learn how to manage tags and deployment issues, and bring analytics data into the environment needed to drive smarter, more informed business decisions.

Contacts.

For questions about this or other Webtrends solutions, contact:

Michael Cook
michael.cook@webtrends.com
(503) 553-2143

or

Abed Farhan
abed.farhan@webtrends.com
(503) 553-2747

Webtrends: Solving Complex Business Challenges for More than 15 Years

Webtrends turns data into understanding of customers and business trends. Its powerful technology and professional services help organizations crunch the numbers from their web sites, blogs and other online and offline systems to gain the insight that feeds business success.

Founder of the web analytics industry, Webtrends offers unmatched experience and leadership across a broad range of online and offline measurement and analysis, including visitor-level and aggregate web site analytics, social media measurement and pay-per-click advertising.

Webtrends technology and services deliver unsurpassed:

- **Power.** Virtually unlimited data collection and multidimensional analysis uncovers the trends and customer behaviors that matter most.
- **Elegance.** An intuitive interface lets anyone slice and dice data and uncover meaningful insights.
- **Openness.** Webtrends' open APIs and support for other industry standards streamline the integration of web and offline data, so you can gain cross-channel perspective on marketing and business trends, as well as create best-of-breed solutions to your business challenges.

Webtrends Technology Turns Data into Understanding

Webtrends offers the enterprise measurement and analytics industry's most powerful technology for the marketing optimization and data management tasks that organizations perform most often: search engine marketing, web site measurement, visitor analysis and social media measurement.

Ad Director. Manage, optimize and profitably expand your paid-search advertising portfolio using automated, self-learning technology.

Analytics. Measure all aspects of your organization's website using full-function, enterprise web analytics.

Marketing Warehouse. Access and connect online, visitor-level data with offline data and systems.

Score. Measure engagement and reveal the visitors who offer the most value to your business.

Visitor Intelligence. Uncover the preferences over time of your individual customers to improve segmentation and targeting.

WebSphere Portal Version 6.1: Your World. Your Way.

WebSphere Portal Version 6.1 is an enterprise portal solution with the complete portal services necessary to deliver a single point of personalized interaction with applications, content, business processes and people. The unified user experience can help you improve overall productivity and customer satisfaction. WebSphere Portal rationalizes and simplifies IT infrastructure by delivering a complete set of portal platform services:

- Framework services.

- Integration services.

- Content services.

- Collaboration services.

These services can be leveraged by WebSphere Portlet Factory Designer to build role-based, portal applications that can help improve the effectiveness of your organization. And because WebSphere Portal utilizes an SOA, these applications can use existing assets and can be quickly modified to meet changing roles and business conditions.

IBM WebSphere Portal Version 6.1 consists of three offerings:

- IBM WebSphere Portal Server.

- IBM WebSphere Portal Enable.

- IBM WebSphere Portal Extend.

Key new features

- Live Text tagging presents 'one-click' access to relevant supporting information that "pops up" on the page while executing a business process.

- Portal REST services further opens the portal platform to composite "mashup" applications with services feeds from other Web applications.

- New Client Side Aggregation reduces server-side processing, and dramatically improves end-user performance.

- Client Side Feed Consumption provides highly efficient integration of information through feeds (Atom and RSS).

- New portal 'Site Wizard' enables self-service creation of independent virtual portals.

- New 'Theme Customizer' portlet delivers wizard-driven branding and in-place customization.

- Updated out-of-the-box Internet and intranet site jumpstart templates offer fast delivery and customization of virtual portal sites.

- New 'One click' Site Management application simplifies control of staging to production capabilities with rollback and versioning.

- Expanded Site Layout support (HTML/PHP), continues to open the platform to additional content sources.

- Support for the very latest portlet open standards (JSR286, WSRP 2.0).

- Web Application Integrator provides a method to easily embed existing Web applications in WebSphere portal for greater value.

- New WebSphere Portlet Factory release delivers an easier out-of-the-box experience combined with new Web 2.0 features and support for visual application development of desktop-like Web applications.

Key benefits

- Deliver exceptional Web experiences. Take advantage of new, nimble Web 2.0 technologies and philosophies that put new business models in reach.

- Extend your business assets. Create and deploy custom-branded, market-driven solutions comprised of existing and net-new business assets, and re-purpose and reuse as appropriate

- Run your business efficiently. Built on WebSphere software so that enterprise-class applications, processes and transactions can reside on a proven, reliable, scalable and high-performance foundation.

- Grow as you go. "Snap on" business-specific capabilities that speed time to value and allow you to pursue new market opportunities – all while reducing the of cost of deploying portals.

The Webtrends Implementation Process

Webtrends offers multiple implementation options. Webtrends On Demand customers should use the SmartSource Data Collection (SDC) method. Webtrends Analytics Software customers can use the SmartSource Data Collection method or the Site Analyzer files that WebSphere produces. Software customers can collect and use both reporting methods. Each approach uses distinct data collection methods and produces different data. SDC captures page views, a page being displayed in a browser. The Site Analyzer reports on hits, requests to the server.

For support of this solutions of trouble shooting, please contact Webtrends Professional Services, for time and materials assistance.

The Webtrends Implementation Process: SmartSource Data Collector

Webtrends uses a JavaScript tag to collect data from your Web site. The script sends basic page and visitor information to Webtrends servers when a page that includes the tag is rendered by the visitor's browser.

Additional data can be passed via the JavaScript tag using META tags or parameters. This method is primarily used to capture page or event-specific information. After the data has been collected, Webtrends analysis engines apply further configuration setting information (supplied via the Administration Console) to generate the final report data for viewing.

The implementation process follows four key steps, which correspond to the key phases of the reporting process described above:

- **Implementation of Webtrends JavaScript.** The Webtrends base tag includes the common JavaScript functions, which are site-independent and allow data collection for most Webtrends Standard Reports. The Webtrends JavaScript Tag is provided by your Webtrends consultant or can be generated by using the Webtrends Tag Builder. For purposes of consistent tracking throughout WebSphere, this tag has been modified in a number of areas. These modifications are detailed later in this document, along with the deployment process for the tag.
- **Modification of Webtrends.js File To Collect WebSphere Specific Data.** Specific tracking functions within WebSphere necessitate changes within the standard Webtrends.js created by the Webtrends Tag Builder. These changes allow for the tracking of various content elements without additional coding. The changes to this file are described in detail later in this document.
- **Modification of WebSphere Pages to Pass Additional Data.** WebSphere pages can be updated to pass data that can allow for richer reporting. This data collection requires some modification of standard JSP pages to send data. These edits are optional.
- **Report Configuration.** Once the Webtrends data capture methods have been deployed, the fourth step in the process is completion of report configuration within the Webtrends Administration interface. Report configuration provides additional specifications for reporting on the data captured by the Webtrends JavaScript and META Tag data collection process. The content of the reports to be configured as part of this project have been specified as part of the Business Solution Design (BSD) document. This document includes additional technical specifications regarding the configuration of these reports which are configured by your Webtrends consultant.

Webtrends Tag Deployment Strategy

Your Webtrends consultant may provide Webtrends JavaScript tags during an engagement, or you can create your own tags using the Webtrends Tag Builder (http://tagbuilder.Webtrends.com). Customers who create their own tag through the use of Tag Builder can refer to the Help section on the Web site.

The configuration of the tag includes these files:

- JavaScript file (Webtrends.js), which includes both customer and WebSphere specific configuration and standard functionality.
- WebSphere Default Skin Page (Default.jsp), which contains a modified Default.jsp page code as installed in WebSphere, as well as the JavaScript code from the Webtrends.js file.
- WebSphere Head Page (head_extras.jspf) which contains a modified head_extras.jspf code, which will collect WebSphere specific data using META tags.

WebSphere Tagging

The key to enabling a rich set of reporting from IBM WebSphere Portal Version 6.1/WSP is the tagging needed for data collection. This is not unlike the activity needed to tag other rich Internet web sites. Much of the action is driven based on Portlets and the interaction they have with the underlying functionality built into WebSphere itself. The tagging process is, therefore, split between two main elements. First there is the Webtrends data collection tag. This tag is basically the standard tag delivered via the Webtrends tag builder application – http://tagbuilder.Webtrends.com. There are a few modifications that have been made to this tag, however, to provide consistency between the reporting of user interactions with WebSphere elements. These changes are optional and highlighted below. The changes allows for the automatic tracking of user names, Search, Portlets, Content Groups and Sub-Groups.

The second part of the tagging process is a set of modifications made to the code supplied by IBM. The changes that can be applied to that file are also detailed below.

WebSphere Architecture

WebSphere Portal is a framework--including a runtime server, services, tools, and many other features--that you can use to integrate your enterprise into a single, customizable interface called a portal. An enterprise portal combines components, applications, processes, and content from a wide variety of sources into a unified presentation, which your users can access from a wide variety of devices.

You can customize the portal based on user or job roles, security needs, device settings, personal preferences, and administrative settings. You can also define workflows to support your business processes. You can manage your portal's content using IBM's Web Content Management, which is integrated with WebSphere Portal. Figure 1 shows an example company portal.
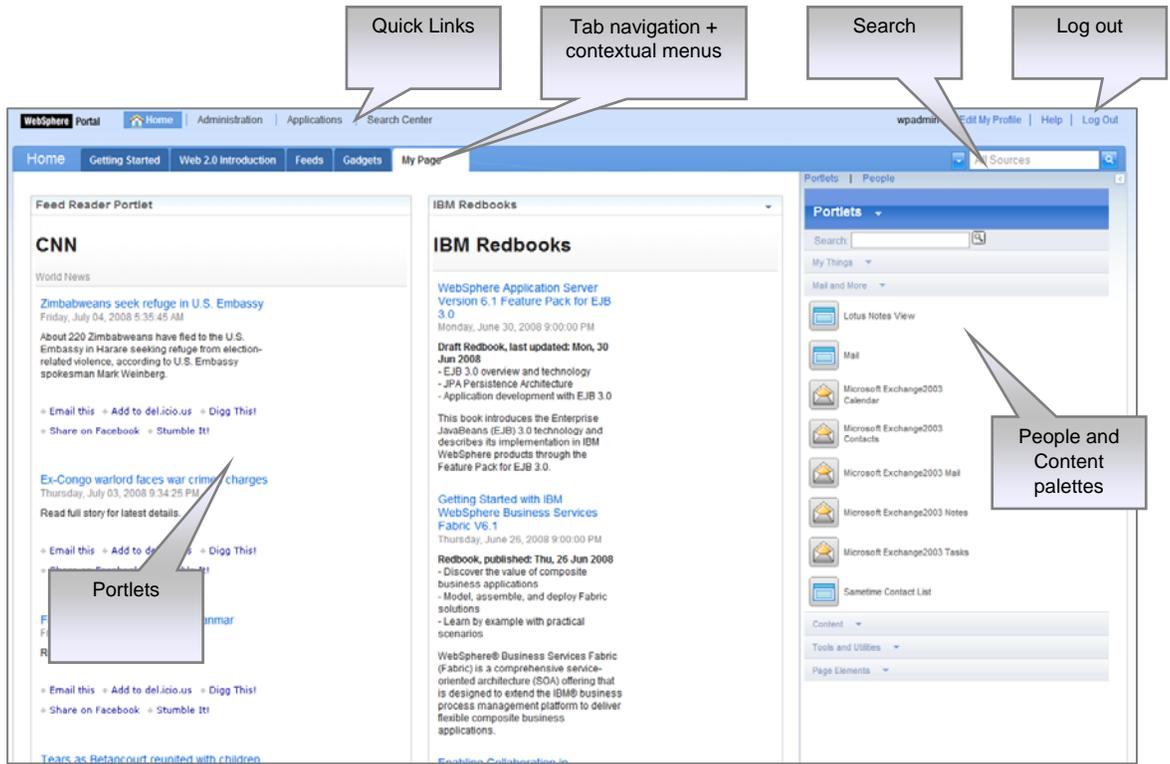
Figure 1. A company portal

The figure below shows the relationship between the Default.jsp presentation skin and how individual content pages implement that layout. The Default.jsp is responsible for governing that general layout and imbeds the placeholders used by content pages.

Figure 2 - WebSphere Architecture

WebSphere Architecture with Webtrends Tag

Tagging a site for Webtrends tracking and reporting needs to keep this architecture in mind and utilize it to the best advantage.

Figure 3 - WebSphere Architecture with Webtrends Tracking Script

The figure above depicts the changes made in a typical WebSphere portal. The Webtrends data collection tag captures standard page level details as well as WSP specific information and sends that data to the data collection server.

Tagging Default.jsp

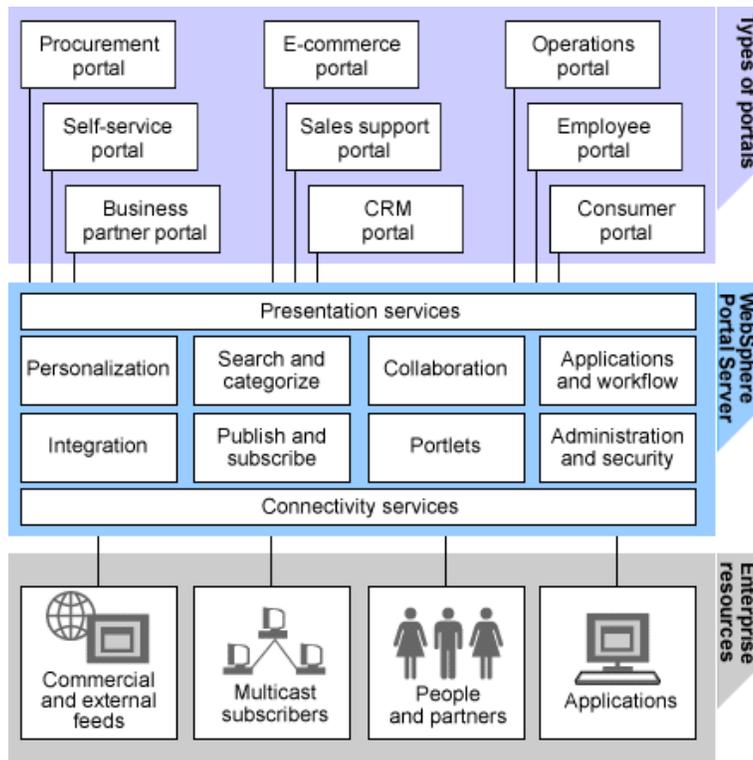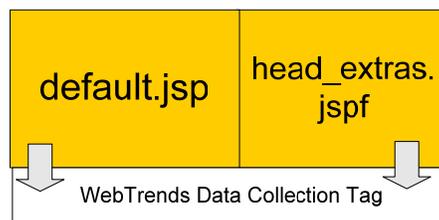The quickest method is to implement Webtrends code is to modify an existing Default.jsp page after the site has been created. This file is used within WebSphere to render pages.

All of these files can be found in the same directory on each server with the WebSphere farm:

(Root)/themes/html/Portal/Default.jsp

The data following this paragraph needs to be added. This modification will create the base for Webtrends tracking. The elements that are bolded should be modified based on your unique information. This code should be added right before the close body tag.

```
<!-- START OF SmartSource Data Collector TAG -->
<!-- Copyright (c) 1996-2009 Webtrends Inc.  All rights reserved. -->
<script src="(LOCATION OF JS FILE) /Webtrends.js" type="text/javascript"></script>
<script type="text/javascript">
//<![CDATA[
var _tag=new Webtrends();
_tag.dcsGetId();
//]]>>
</script>
<script type="text/javascript">
//<![CDATA[
// Add custom parameters here.
//_tag.DCSext.param_name=param_value;
_tag.dcsCollect();
//]]>>
</script>
<noscript>
<div><img alt="DCSIMG" id="DCSIMG" width="1" height="1"
src="http://(INSERT SDC LOCATION HERE)/(INSERT DCS ID
HERE)/njs.gif?dcsuri=/nojavascript&amp;WT.js=No&amp;WT.tv=8.6.2"/></div>
</noscript>
<!-- END OF SmartSource Data Collector TAG -->
```

Tagging head_extras.jspf

WebSphere builds the head of the document using this file. This makes the file the correct place to pass data to Webtrends using the standard META tags. This code should be added right at the bottom of the file.

If you are testing the modifications to this page remember to save an updated version of default.jsp. This page will be cached by the WebSphere platform unless a change to the defualt.jsp file has been recorded.

URI Overwrite head_extras.jspf

The first optional addition is to overwrite the standard URI string produced by a website that has not created a site map. This means their website URL may have "unfriendly URLs".

An example of an "unfriendly URL" would be:
http://www.ibm.com/wps/portal/!ut/p/kcxml/04_Sj9SPykssy0xPLMnMz0vM0Y_QjzKLd4w3dnQESYGYRq6m-pEoYgbxjggRX4_83FT9oLQifW_9AP2C3NCIckdHRQDg-gGH/delta/base64xml/L0lKWWttUSEhL3dITUFDc0FFVUFObby80SUVhREFBIS9lbg!

These URLs may be hard to understand for anyone looking at standard Webtrends pages reports. Alternately, the URI overwrite option allows us to replace the URL string with something that is more readable. For Example:

http://www.ibm.com/My Website/The Main Area/News and Views/

This option uses the site structure created on the WebSphere backend to populate the URI automatically. No additional modifications are required.

The following code needs to be added:

<!--URI Overwrite-->

```
<META NAME="DCS.dcsuri" CONTENT="/<c:out value="${siteTitle}"/>/<c-rt:if
test="${!renderPlaceholdersOnly}"><%@ page
import="com.ibm.portal.model.NavigationSelectionModelProvider" %><%@ page
import="com.ibm.portal.navigation.NavigationSelectionModel" %><%@ page
import="com.ibm.portal.navigation.NavigationNode" %><%@ page
import="com.ibm.portal.content.ContentNodeType" %><%@ page import="java.util.Iterator" %><%if
(navigationSelectionModelHome != null){ NavigationSelectionModelProvider provider =
navigationSelectionModelHome.getNavigationSelectionModelProvider();NavigationSelectionModel model =
provider.getNavigationSelectionModel(request, response);java.util.Iterator iterator = model.iterator();int
startLevel = themePolicy.getBreadCrumbStartLevel();int maxItems =
themePolicy.getBreadCrumbMaxLevels();int currLevel = 0;int numItems = 0;while (iterator.hasNext() &&
numItems < maxItems){NavigationNode node = (NavigationNode) iterator.next();ContentNodeType
nodeType = node.getContentNode().getContentNodeType();%><c-rt:if test="<%=currLevel >= startLevel
%>"><c:set var="crumbtrailItemTitle"><portal-fmt:title varname="<%=node%>"/></c:set><c-
rt:choose><c-rt:when test="<%= nodeType.equals(ContentNodeType.LABEL) %>"><% if (numItems > 0)
{ %>/<% } %><c:out value='${crumbtrailItemTitle}' escapeXml='true' /><% numItems ++; %></c-
rt:when><c-rt:otherwise><% if (numItems > 0) { %>/<% } %><c:out value='${crumbtrailItemTitle}'
escapeXml='true' /><% numItems ++; %></c-rt:otherwise></c-rt:choose></c-rt:if><%  currLevel ++;
}}%></c-rt:if>" />
```

<!--End URI Overwrite-->

Breadcrumb head_extras.jspf

The second optional addition is to create a breadcrumb report, which will enable users to create a Webtrends report that will report on activity using the WebSphere Portal navigation structure. Again, this option simply utilizes information that is already available.

The code that needs to be added is as follows:

<!--Breadcrumb-->

```
<META NAME="WT.wsp_bc" CONTENT="<c:out value="${siteTitle}"/>|<c-rt:if
test="${!renderPlaceholdersOnly}"><%@ page
import="com.ibm.portal.model.NavigationSelectionModelProvider" %><%@ page
import="com.ibm.portal.navigation.NavigationSelectionModel" %><%@ page
import="com.ibm.portal.navigation.NavigationNode" %><%@ page
import="com.ibm.portal.content.ContentNodeType" %><%@ page import="java.util.Iterator" %><%if
(navigationSelectionModelHome != null){ NavigationSelectionModelProvider provider =
navigationSelectionModelHome.getNavigationSelectionModelProvider();NavigationSelectionModel model =
provider.getNavigationSelectionModel(request, response);java.util.Iterator iterator = model.iterator();int
startLevel = themePolicy.getBreadCrumbStartLevel();int maxItems =
themePolicy.getBreadCrumbMaxLevels();int currLevel = 0;int numItems = 0;while (iterator.hasNext() &&
numItems < maxItems){NavigationNode node = (NavigationNode) iterator.next();ContentNodeType
nodeType = node.getContentNode().getContentNodeType();%><c-rt:if test="<%=currLevel >= startLevel
%>"><c:set var="crumbtrailItemTitle"><portal-fmt:title varname="<%=node%>"/></c:set><c-
rt:choose><c-rt:when test="<%= nodeType.equals(ContentNodeType.LABEL) %>"><% if (numItems > 0)
{ %>|<% } %><c:out value='${crumbtrailItemTitle}' escapeXml='true' /><% numItems ++; %></c-
rt:when><c-rt:otherwise><% if (numItems > 0) { %>|<% } %><c:out value='${crumbtrailItemTitle}'
escapeXml='true' /><% numItems ++; %></c-rt:otherwise></c-rt:choose></c-rt:if><% currLevel ++;
}}%></c-rt:if>" />
```

Logged In User Name head_extras.jspf

In a WebSphere environment it is normal for users to log into the WebSphere server in order to gain access.
This log in also determines their level of access as well. WebSphere populates a JSP variable that we can
access. This code block will check to see if the user is logged in. If so, the username will be sent to
Webtrends for reporting.

The code that needs to be added is as follows:

```
<!--User Tracking-->

<portal-logic:if loggedIn="yes"><META NAME="DCS.dcsaut" CONTENT="<portal-fmt:user attribute="cn"/>"
/></portal-logic:if>
```

Changes to the Standard Webtrends Data Collection Tag

There are a few changes that are made to the standard Webtrends data collection tag as it is built in the tag
builder at http://tagbuilder.Webtrends.com. These changes do a number of things:

- Capture values for Content Group and Content Sub-group based on the WSP Breadcrumb
- Capture portlet information
- Capture search results and number of results recorded
- Add additional data to the URI overwrite functionality

Adding The Core Functionality To Webtrends.js

Enabling WSP specific tracking using Webtrends requires two modifications to the
existing code stream. The bolded code listed below is the full addition we need to
make. Make sure to insert this code in the proper place. Once the changes are made
the code should be placed on the server. (Root)/themes/html/Portal/js/ would be a
good location to place the file.

Webtrends.prototype.dcsWSPall=function(){

```
        var elems;
        var doURIH1=true;
        var doSRC=true;
        var doportlets=true;
        var docg=true;

        if (doURIH1){
                var h1Collection = document.getElementsByTagName("h1");
                for (var i=0; i<h1Collection.length; i++) {
                H1URI = document.getElementsByTagName("h1")[0].childNodes[0].nodeValue;
                _tag.DCS.dcsuri = _tag.DCS.dcsuri + "/" + H1URI;
                }
                }


//MC Search Code
        if (doSRC){
                SearchTest =
document.getElementById("ibm_portal_search_widgets_ResultsFetcher_0");
                if (SearchTest!= null){
                var SK = "[\\?&]query=([^&#]*)";
                var SK = new RegExp( SK );
                osst = SK.exec(window.location.href);
                SRC_pull_final = "0";
                var strongCollection = document.getElementsByTagName("strong");
                for (var i=0; i<strongCollection.length; i++) {
                SRC_pull =
document.getElementsByTagName("strong")[i].childNodes[0].nodeValue;
                if (osst[1]== SRC_pull){
                var i=i-2;
                SRC_pull_final =
document.getElementsByTagName("strong")[i].childNodes[0].nodeValue;
                var i=i+2;
                }
                this.WT.oss_r = SRC_pull_final;
                this.WT.oss = osst[1];
                }
                }
                }
//End MC Search Code

//MC pt Code
                if (doportlets){
                if (window.location.search){
                var PTmatch = /WT..ortlet/.test(window.location.search);
                }
                if (PTmatch == true) {
                var PT = "[\\?&]WT..ortlet=([^&#]*)";
                var PT = new RegExp( PT );
                prtl = PT.exec(window.location.search);
                pt_final = prtl[1] + ";";
                pt_a = "Click;";
```

```
            }
            else {
            pt_final = "";
            pt_a = "";
            }
            }


            if (doportlets){
            var divCollection = document.getElementsByTagName("div");
            for (var i=0; i<divCollection.length; i++) {
            if (divCollection[i].className == "BoxHeader")
            {
            pt_pull =
document.getElementsByTagName("div")[i].childNodes[0].nodeValue;
            pt_final = pt_final + pt_pull + ";";
            pt_a = pt_a +"View" +";";
                            }
            }
            this.WT.portlet = pt_final;
            this.WT.portlet_a = pt_a;
            }
//End MC pt Code


//Start cg
        if (docg){
        <!--Checks to see if breadcrumb value is available, if so splits value-->
        if (typeof(_tag.WT.wsp_bc)!="undefined"){
        var tempcgs_array = _tag.WT.wsp_bc.split("|");

        <!--Sets CG if no CG is set and breadcrumb value is available-->
        if (typeof(_tag.WT.cg_n)=="undefined" || tempcgs_array[1]!="undefined") {
        this.WT.cg_n =  tempcgs_array[1];
        }

        <!--Sets SCG if no SCG is set and breadcrumb value is available-->
        if (typeof(_tag.WT.cg_s)=="undefined"  || tempcgs_array[2]!="undefined") {
        this.WT.cg_s=tempcgs_array[2];
        }
        }
        }
//End cg

        }

Webtrends.prototype.dcsCollect=function(){
   if (this.enabled){
        this.dcsVar();
        this.dcsMETA();
        <!--WSP Code-->
        this.dcsWSPall();
        <!--WSP Code-->
```

```
this.dcsAdv();
this.dcsTag();
                }
                }
```

The Collection Switches

The JavaScript code has a number of switches that can be enabled or disabled. By default these items are enabled. However, they can be disabled by changing the "true" value to "false".

URI Enhancement (var doURIH1=true;)
This switch allows you to add additional information to the URI overwrite stream by adding another level to the URI, if it is available. If you are using the URI overwrite in the head_extras.jspf file, this should be enabled. If you are not, this should be disabled. By default this code is searching for the results contained inside of the <H1> HTML tag. If you require a modified version of this functionality, please contact Webtrends Professional Services for JavaScript functions tailored to your unique skin.

Onsite Search (var doSRC=true;)
This switch will allow you to automatically track search results and number of result returned. If you are using a modified version of the WSP skins or templates you may want to test the functionality of the number of results returned to ensure the proper number is being set. By default this code is searching for the results contained inside of the **<Strong>** HTML tags, which are used to bold the page content. If you require a modified version of this functionality, please contact Webtrends Professional Services for JavaScript functions tailored to your unique skin.

Portlet View Tracking (var doportlets=true;)
This switch will allow you to track the portlets displayed on a page. If you are using a modified version of the WSP skins or templates you may want to test the functionality to ensure the proper data is being set. By default this code is searching for the text contained inside of a DIV with a class named "BoxHeader". If you are using a different class name for your portlet titles, you should change "BoxHeader" in the JavaScript to reflect the name of your custom class. If you require a modified version of this functionality, please contact Webtrends Professional Services for JavaScript functions tailored to your unique skin.

Content Group and Content Sub-group (var docg=true;)
Content Groups allow us to report on content groupings at an aggregate level. For example, I may want to know how much time visitors spent in the News section as a whole rather than each individual page that was visited. The code automatically passes this data to the Webtrends tag using the Breadcrumb functionality in the head_extras.jspf file. If you have not enabled that code you cannot use this option.

Tag Function for Common WebSphere Data Elements

While the code provided in this document will allow for the capture of many standard elements, developers can pass this data to the Webtrends tag using their own preferred method. These parameters should be placed in the document head using META tags using the following format:

<META NAME="**WEBTRENDS PARAMETER**" CONTENT="**INSERT PARAMETER VALUE HERE**">.

The bolded content should be updated to reflect the data collection goal. The standard data collection parameters are:

● Data Element	● Webtrends Parameter Populated

● User Name	DCS.dcsaut
● Content Group	WT.cg_n
● Content Subgroup	WT.cg_s
● On Site Search Term	WT.oss
● On Site Search Results	WT.oss_r
● Navigation Breadcrumb (levels should be delimited using a pipe character "|")	WT.wsp_bc
● Portlets (portlets should be delimited using a semicolon ";")	WT.portlet
● Portlet Action (required if using Portlet) This parameter should have a value of "view" or "click" depending on the context"	WT.portlet_a
● Alternate URI string	DCS.dcsuri

Tracking Portlet Clicks

If you wish to track clickthrough on a portlet link, the following parameter should be added to the end of the URL ?WT.portlet=**(PORTLET NAME)**. "?" should be changed to "&" if you already have a "?" in the URL string. The name of the portlet should be consistent with the title the portlet uses on the page it is displayed on to ensure consistent reporting in Webtrends. If you are using a portlet that has clicks that do not result in a page change (e.g., an AJAX application) you should use the dcsMultiTrack functionality (see Appendix A) to capture that interaction. In addition to the standard Webtrends parameter, you will want to capture the "WT.portlet" and "WT.portlet_a" parameters as detailed in the previous table.

Summary of Steps Required for Tag Deployment

1. Configure new tag using Tag Builder

2. Update default.jsp and head_extras.jspf

3. Update Webtrends.js file

4. Place update files on server (Root)/themes/html/Portal/

5. Additional manual configuration of tag if required

6. Test deployment of tags within development environment

7. Promote changes to production environment

8. Verify data collection at Webtrends

Administrative Configuration

WebSphere Specific Reports

There are a number of reports that have been designed based on the data captured on the updated JavaScript code and Webtrends JavaScript tag. Among these are the following:

- Activity by User
- Breadcrumbs
- Portlets Viewed
- Portlets Clicked
- Portlets Viewed by Page

Activity by User

The Activity by User report provides a list of the most active users in the organization. For each user, the report displays an overall set of metrics. The standard set of metrics in this report are:

- Visits
- Page Views
- Visit Page Views
- Visit Duration (Minutes)

User Activities

User	Visits▼	Page Views	Views/Visit	Visit Duration (Minutes)
1. Michael Cook	2	34	17	45
2. Michael Love	1	2	2	2

Figure 4 - Activity by User Report

The following is the setup information for the Activity by User Report.

Name: WebSphere Activity User
Locked: No
Type: Standard Custom Report
Category: WebSphere
Report Title: Activity by User
Description: Report shows overall measure of key interactions at the user level.

Dimension 1: Authenticated User Name
Dimension 2: None
Measure 1: Visits **(Sum)** - sortable, interval data available
Measure 2: Page Views **(Sum)** - sortable, interval data available
Measure 3: Visit Page Views **(Sum)** - sortable, interval data available
Measure 4: Visit Duration (min) **(Sum)** - sortable, interval data available
**Filters:** None

Accumulation Type: End of Visit (increases memory requirement)
"Visit Page Views" is an end of visit measure.
"Visit Duration (min)" is an end of visit measure.

Breadcrumbs

The Breadcrumbs report displays a list of the various breadcrumbs that are generated as a user navigates into a WebSphere site and sub site. The report dimension is the breadcrumb. This dimension is a drilldown. Each level in the drilldown defines another level to which visitors have navigated during the visit. Navigation in WebSphere back to the "top" or "home" level will, in most cases, begin a new breadcrumb. The metric in the report is the number of occurrences where this breadcrumb is unique.
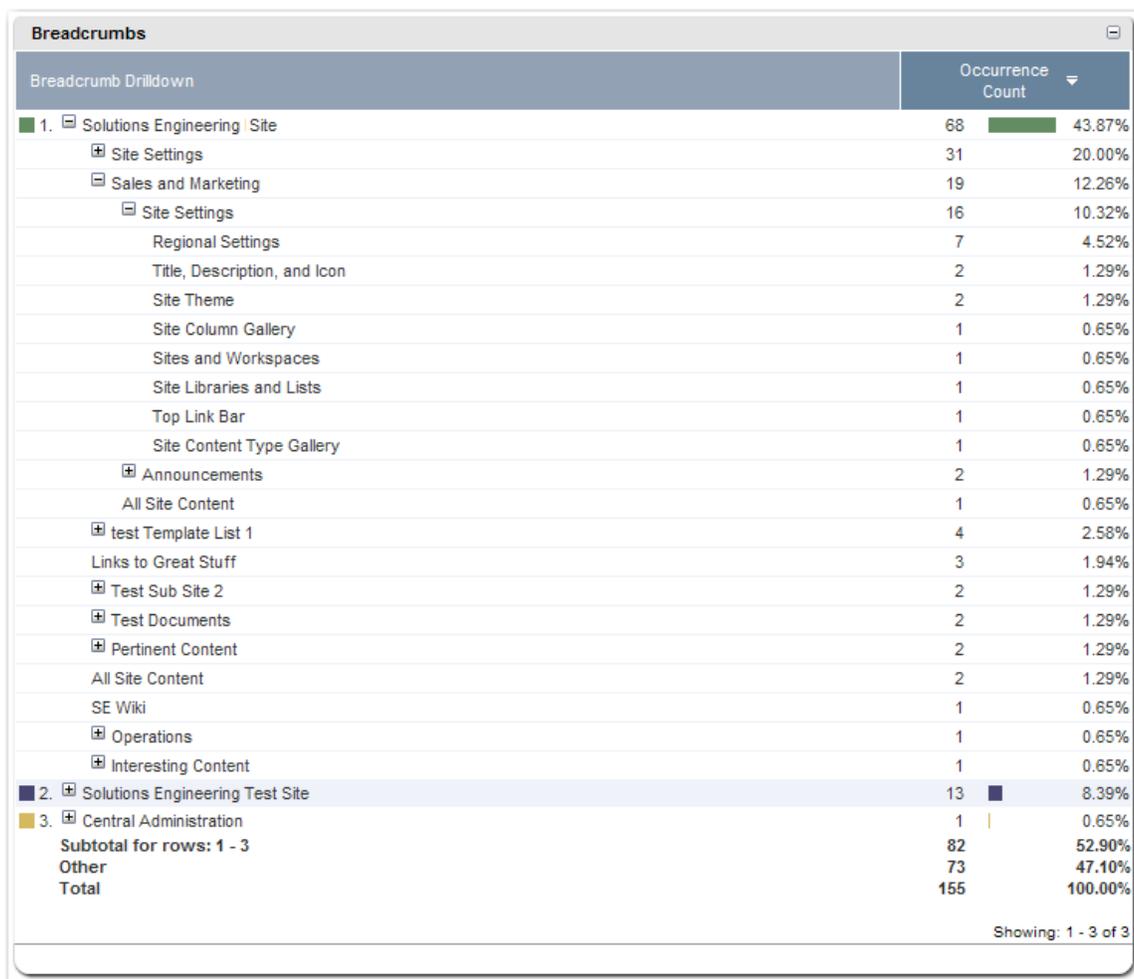
Breadcrumb Drilldown	Occurrence Count	
1. Solutions Engineering Site	68	43.87%
Site Settings	31	20.00%
Sales and Marketing	19	12.26%
Site Settings	16	10.32%
Regional Settings	7	4.52%
Title, Description, and Icon	2	1.29%
Site Theme	2	1.29%
Site Column Gallery	1	0.65%
Sites and Workspaces	1	0.65%
Site Libraries and Lists	1	0.65%
Top Link Bar	1	0.65%
Site Content Type Gallery	1	0.65%
Announcements	2	1.29%
All Site Content	1	0.65%
test Template List 1	4	2.58%
Links to Great Stuff	3	1.94%
Test Sub Site 2	2	1.29%
Test Documents	2	1.29%
Pertinent Content	2	1.29%
All Site Content	2	1.29%
SE Wiki	1	0.65%
Operations	1	0.65%
Interesting Content	1	0.65%
2. Solutions Engineering Test Site	13	8.39%
3. Central Administration	1	0.65%
Subtotal for rows: 1 - 3	**82**	**52.90%**
Other	**73**	**47.10%**
Total	**155**	**100.00%**

Showing: 1 - 3 of 3

Figure 5 - Breadcrumbs Report

The following is the setup information for the Breadcrumbs Report.

        Name: WebSphere Breadcrumb Report
        Locked: No
        Type: Standard Custom Report
        Category: WebSphere

Report Title:	Breadcrumbs
Description:	Breadcrumb reporting showing how overall navigation is conducted at the site level.
Dimension 1:	WebSphere Breadcrumb Drilldown
Dimension 2:	None
Measure 1:	WebSphere Breadcrumb Occurrences **(Count)** - not sortable, interval data not available
Filters:	None
Accumulation Type:	On Event

Portlets Viewed

The Portlets Viewed report displays a list of all Portlets by name that are viewed in the report period. The metric shown is the number of times a portlet was viewed on any page.

Figure 6 - Portlets Viewed Report

The following is the setup information for the Portlets Viewed Report.

Name:	WebSphere Portlets Viewed Report
Locked:	No
Type:	Standard Custom Report
Category:	WebSphere
Report Title:	Portlets Viewed
Description:	Report shows how often different Portlets are shown to users.
Dimension 1:	WebSphere Portlets
Dimension 2:	None
Measure 1:	WebSphere Portlets Views **(Count)** - not sortable, interval data not available
Filters:	None
Accumulation Type:	On Event
Visitor History Dependency:	No Dependencies
Express Analysis Compatibility:	Compatible

Portlets Clicked

The Portlets Clicked report displays a list of all Portlets by name that are Clicked on in the report period. The metric shown is the number of times a portlet was clicked.

WebSphere Portlets Clicked	
WebSphere Portlets	Clicks
1. Links	2
2. Employees On Demand	1

Figure 7 - Portlets Clicked Report

The following is the setup information for the Portlets Viewed Report.

Name:	WebSphere Portlets Clicked Report
Locked:	No
Type:	Standard Custom Report
Category:	WebSphere
Report Title:	Portlets Clicked
Description:	Report shows how often different Portlets are shown to users.
Dimension 1:	WebSphere Portlets
Dimension 2:	None
Measure 1:	WebSphere Portlets Clicked **(Count)** - not sortable, interval data not available
Filters:	None
Accumulation Type:	On Event
Visitor History Dependency:	No Dependencies
Express Analysis Compatibility:	Compatible

Portlets Viewed By Pages

The Portlets Viewed report displays a list of all Portlets by name and which page they were viewed on in the report period. The metric shown is the number of times a portlet was viewed on a specific page.

Figure 8 - Portlets Viewed By Page Report

The following is the setup information for the Portlets Viewed Report.

Name:	WebSphere Portlets Viewed Report
Locked:	No
Type:	Standard Custom Report
Category:	WebSphere
Report Title:	Portlets Viewed
Description:	Report shows how often different Portlets are shown to users.
Dimension 1:	Page Title
Dimension 2:	WebSphere Portlets
Measure 1:	Visits
Measure 2:	% of Visits
Measure 3:	WebSphere Portlets Views **(Count)** - not sortable, interval data not available
Measure 4:	Average Time Viewed

Custom Report Dimensions

There are a number of custom dimensions that have been defined for use in the custom reports that are shown above. These dimensions are:

- WebSphere Breadcrumb Drilldown
- WebSphere Portlets

The sections that follow provide the setup information for each of these dimensions.

WebSphere Breadcrumb Drilldown

Name:	WebSphere Breadcrumb Drilldown

Category:	WebSphere
Locked:	No
Dimension Type:	Drilldown
Based On:	Query Parameter: "WT.wsp_bc"
When to Collect Data:	All Hits
Multiple Values:	No
Drilldown Values:	Yes: delimiter = "|"
Accumulation Type:	On Event
Visitor History Dependency:	Not Dependent
Express Analysis Compatibility:	Not Compatible: Dimension contains drilldown data.

WebSphere Portlets

Name:	WebSphere Portlets
Category:	WebSphere
Locked:	No
Dimension Type:	Standard
Based On:	Query Parameter: "WT.portlet "
When to Collect Data:	All Hits
Multiple Values:	Yes: delimiter = ";"
Drilldown Values:	No
Accumulation Type:	On Event
Visitor History Dependency:	Not Dependent
Express Analysis Compatibility:	Compatible

Custom Report Measures

There are a number of custom measures that have been defined for use in the custom reports that are shown above. These measures are:

- WebSphere Breadcrumb Occurrences
- WebSphere Portlets Views
- WebSphere Portlets Clicks

The sections that follow provide the setup information for each of these measures.

WebSphere Breadcrumb Occurrences

Name:	WebSphere Breadcrumb Occurrences
Category:	WebSphere
Locked:	No
What to Measure:	Query Parameter: "WT.wsp_bc"
When to Measure:	All Hits
Multiple Values:	No
Accumulation Type:	On Event
Visitor History Dependency:	Not Dependent
Express Analysis Compatibility:	Compatible

WebSphere Portlet Views

Name:	WebSphere Portlet Views
Category:	WebSphere
Locked:	No
What to Measure:	Query Parameter: "WT.portlet_a "
When to Measure:	Specific Hit.
	Regular Expression = .*
	URL Parameter | WT.portlet_a = view
Multiple Values:	No
Accumulation Type:	On Event
Visitor History Dependency:	Not Dependent
Express Analysis Compatibility:	**Not Compatible:**
	When to measure is not *all hits*.

WebSphere Portlet Clicks

Name:	WebSphere Portlet Views
Category:	WebSphere
Locked:	No
What to Measure:	Query Parameter: "WT.portlet_a "
When to Measure:	Specific Hit.
	Regular Expression = .*
	URL Parameter | WT.portlet_a = click
Multiple Values:	No
Accumulation Type:	On Event
Visitor History Dependency:	Not Dependent
Express Analysis Compatibility:	**Not Compatible:**
	When to measure is not *all hits*.

The Webtrends Implementation Process: Site Analyzer Log Files

WebSphere Portal implements a logging function for your usage data. The portal writes usage records to a dedicated log file if site analysis logging is enabled. Multiple types of Site Analyzer loggers allow portal administrators to collect statistical data in various areas. The portal server manages the collection of data on its own, but from a business point of view you can also log custom details of business events.

The portal configuration service SiteAnalyzerLogService determines the type of site analysis data that the portal logs at runtime. Depending on the service configuration, the portal logs the following events:

- Management, such as creating, reading, updating, deleting pages
- User management actions, such as creating, reading, updating, and deleting users and groups.
- Page and Portlet views
- Authenticated User Reporting
- Session activities, such as login, logout, time out, login failed

Site Analyzer logs need to be enabled using the WebSphere Administration Console (information at: http://publib.boulder.ibm.com/infocenter/wpdoc/v6r1m0/topic/com.ibm.wp.ent.doc_v6101/admin/ad setcfg.html). Once in the Administration Console, logging must be enabled (information at: http://publib.boulder.ibm.com/infocenter/wpdoc/v6r1m0/topic/com.ibm.wp.ent.doc_v6101/admin/ad saconf.html).

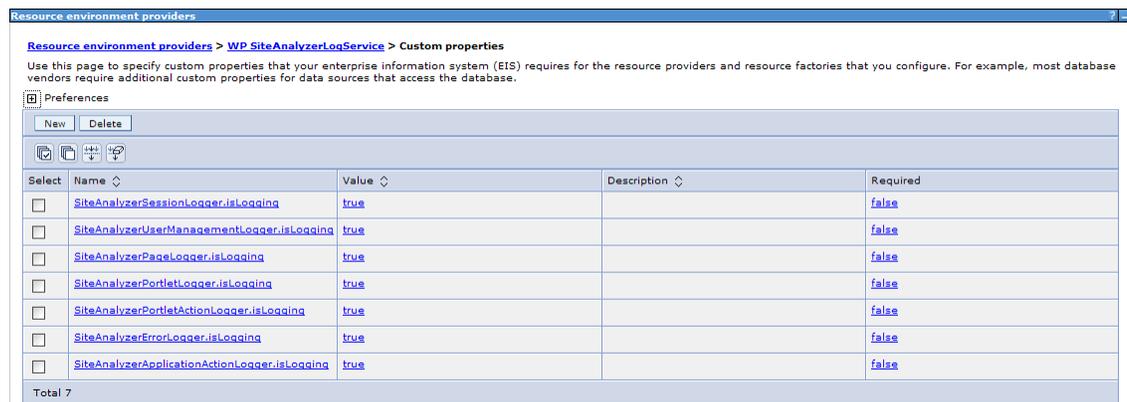| Select	Name ◇	Value ◇	Description ◇	Required
☐	SiteAnalyzerSessionLogger.isLogging	true		false
☐	SiteAnalyzerUserManagementLogger.isLogging	true		false
☐	SiteAnalyzerPageLogger.isLogging	true		false
☐	SiteAnalyzerPortletLogger.isLogging	true		false
☐	SiteAnalyzerPortletActionLogger.isLogging	true		false
☐	SiteAnalyzerErrorLogger.isLogging	true		false
☐	SiteAnalyzerApplicationActionLogger.isLogging	true		false

Total 7

Figure 9 – Example of configuration Screen

To analyze the logs, you simply create a data source that uses the sa(date).log file(s).

Using this simple configuration, you will be presented with a number of standard log file-based Webtrends reports. You can enhance this reporting with Webtrends custom reports.

Administrative Configuration

Tracking User Sessions

Within Site Analyzer, you have two options for tracking users: Authenticated Username and JSESSIONID. Authenticated username is automatically captured in the log file if you have a website where users have to login to access content.

To use this method, access Session Tracking in the Webtrends console. Select Authenticated User Name as the method and save, and then apply this tracking method to your Webtrends profile.

Figure 10 – Example of configuration Screen

The Authenticated Usernames report in Webtrends will automatically report on this field. Additionally, the New vs. Returning Visitor report will be populated with data.
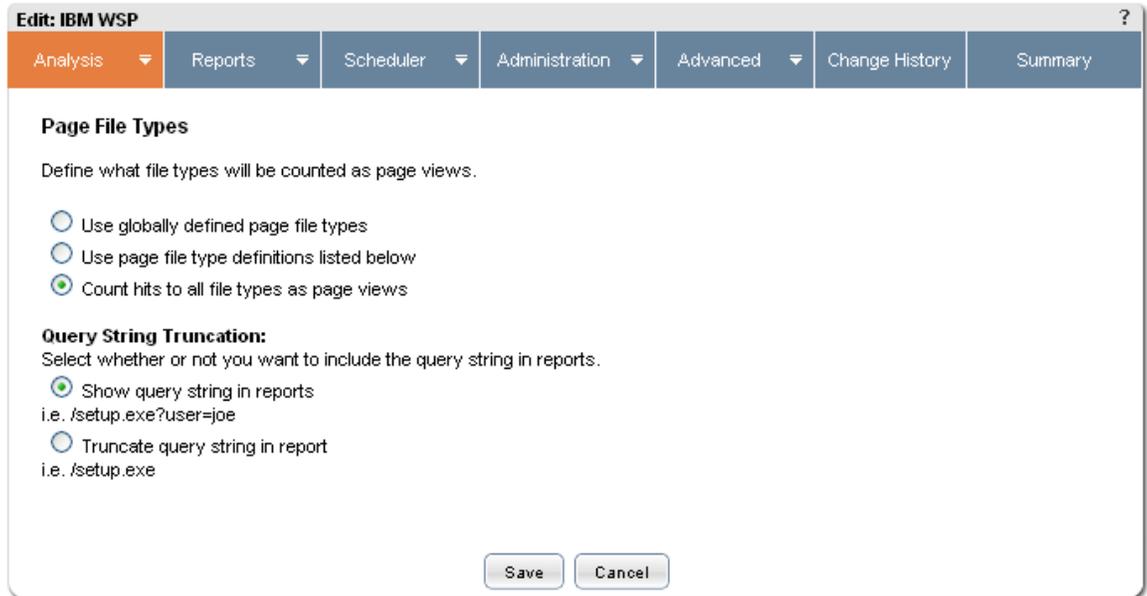

Figure 11 – Authenticated Usernames report

A JSESSIONID session cookie is created when any user visits the site. The drawback of using this method for tracking is the session resets every time a visitor comes back to the website after 30 minutes of inactivity. This makes it impossible to track New vs. Returning.

 To use this method, access Session Tracking in the Webtrends console. Select Cookie as the method, enter the value "JSESSIONID" and save, and then apply this tracking method to your Webtrends profile.

Configuration of Page Types

Because IBM WebSphere produces filenames that may not match standard file name conventions, we recommend turning on "Count hits to all file types as page views" in the WebTrends profile. This will ensure no hits are lost.

Figure 12 - Authenticated Usernames report

WebSphere Specific Custom Reports

There are a number of reports that have been designed based on the data captured on the updated JavaScript code and WebTrends JavaScript tag. Among these are the following:

- Commands
- New vs. Returning Visitors
- Page Customizer
- Pages Report
- Portlet Report
- User Management

Commands Report

The Commands report provides data on any user management events that are made through the portal's administrative interface. This includes Logins, Logouts, User Management, Customizer and Group Management. This report can be enhanced further by adding Authenticated Username as a primary dimensions, allowing you to see what users are making these changes.

Figure 13 -Commands report

The following is the setup information for the Activity by User Report.

Name:	Commands
Locked:	No
Type:	Standard Custom Report
Category:	WebSphere
Report Title:	Commands
Description:	Provides data on any command events that are made through the portal's administrative interface
Dimension 1:	Command
Dimension 2:	None
Measure 1:	Hits - sortable, interval data available
Filters:	None
Accumulation Type:	On Event

Page Customizer

The Page Customizer provides data on any use of the Page Customizer to edit, create or delete pages using the portal's administrative interface. If the page has been assigned a name, this will also appear in the report.

Figure 14 - Page Customizer report

The following is the setup information for the Page Customizer Report.

Name:	Page Customizer
Locked:	No
Type:	Standard Custom Report
Category:	WebSphere
Report Title:	Page Customizer
Description:	Provides data on any page customization events that are made through the portal's administrative interface
Dimension 1:	Page Customized
Dimension 2:	Page Name
Measure 1:	Hits - sortable, interval data available
Filters:	None
Accumulation Type:	On Event

Pages Report

The Pages Report is an alternative to the standard "out-of-the-box" pages report. The standard pages report will include all events, including portlet actions. This modified version will provide data only related to pages hits.

Pages Report		Items Per Page: 50	
IBM Pages		**Hits** ▼	**Visits**
■ 1.	http://localhost/Page/6_CGAH47L00003E02TNFNSMN3G H7/null/	45	1
■ 2.	http://localhost/Page/6_CGAH47L000HO702TUI0ESH3O 92/null/	31	8
■ 3.	http://localhost/Page/Department Template/admin/	24	1
■ 4.	http://localhost/Page/Department Template/new/news/	19	2
■ 5.	http://localhost/Page/6_CGAH47L00003E02TNFNSMN3G R1/null/	18	8
■ 6.	http://localhost/Page/Department Template/docs/	12	3
■ 7.	http://localhost/Page/6_CGAH47L00003E02TNFNSMN3G P6/null/	9	1

Figure 15 -Pages Report

The following is the setup information for the Pages Report.

Name:	Pages Report
Locked:	No
Type:	Standard Custom Report
Category:	WebSphere
Report Title:	Pages Report
Description:	Provides filter data on just the pages hits by visitors

Dimension 1:	IBM Pages
Dimension 2:	None
Measure 1:	Hits - sortable, interval data available
Filters:	Page filter (Include)
Accumulation Type:	On Event

Portlet Report

The Portlet Report provides data on portlets accessed as well as the portlet id, the mode and the state the portlet was in during the hit.

Figure 16 - Portlet Report

The following is the setup information for the Portlet Report.

Name:	Portlet Report
Locked:	No
Type:	Standard Custom Report
Category:	WebSphere
Report Title:	Portlet Report
Description:	Provides data on portlets accessed, the mode and state
Dimension 1:	Portlet Name
Dimension 2:	Portlet Drilldown
Measure 1:	Hits - sortable, interval data available
Measure 2:	Visits - sortable, interval data available

<div align="center">

Filters: None

Accumulation Type: On Event

</div>

User Management

The User Management report provides data on any user management events that are made through the portal's administrative interface. This report can be enhanced further by adding Authenticated Username as a primary dimensions, allowing you to see what users are making these changes.

Figure 17 - User Management report

The following is the setup information for the User Management Report.
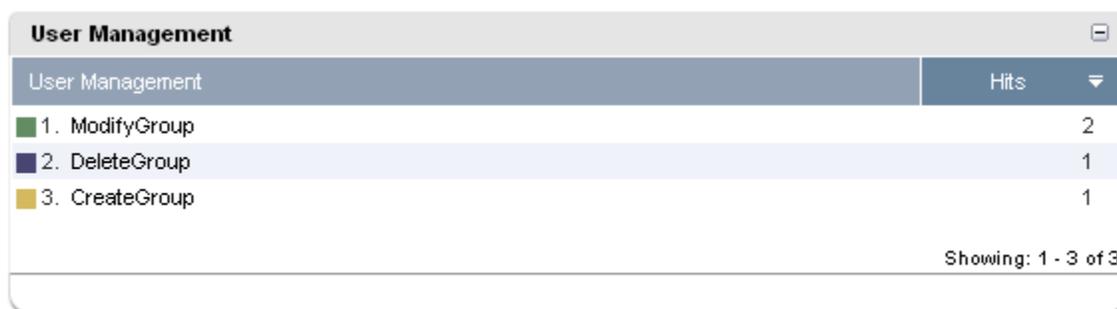
<div align="center">

Name: User Management

Locked: No

Type: Standard Custom Report

Category: WebSphere

Report Title: User Management

Description: Provides data on any user management events that are made through the portal's administrative interface

Dimension 1: User Management

Dimension 2: None

Measure 1: Hits - sortable, interval data available

Filters: None

Accumulation Type: On Event

</div>

Custom Report Dimensions

There are a number of custom dimensions that have been defined for use in the custom reports that are shown above. These dimensions are:

- Page Object ID
- Page Name

Page **32** of **41**

- Portlet Mode
- Portlet State
- Portlet Drilldown
- Command
- Portlet Name
- User Management
- Page Customizer
- IBM Pages

The sections that follow provide the setup information for each of these dimensions.

Page Object ID

Name:	Page Object ID
Category:	WebSphere
Locked:	No
Dimension Type:	Standard
Based On:	Query Parameter: "Page"
When to Collect Data:	Hits that match the regex ".*Command/Customizer.*"
Multiple Values:	No
Drilldown Values:	No
Accumulation Type:	On Event
Visitor History Dependency:	Not Dependent
Express Analysis Compatibility:	Not Compatible: Dimension uses regular expression for when to collect

Page Name

Name:	Page Name
Category:	WebSphere
Locked:	No
Dimension Type:	Standard
Based On:	URL with Query String
When to Collect Data:	Hits that match the regex ".*Command/Customizer.*"
Accumulation Type:	On Event
Visitor History Dependency:	Not Dependent
Express Analysis Compatibility:	Not Compatible: Dimension based on URL with Query String.

Portlet Mode

Name:	Portlet Mode
Category:	WebSphere
Locked:	No
Dimension Type:	Standard

Based On:	Query Parameter: "PortletMode"
When to Collect Data:	Hits that match the regex "PortletPID" is equal to ".*"
Multiple Values:	No
Drilldown Values:	No
Accumulation Type:	On Event
Visitor History Dependency:	Not Dependent
Express Analysis Compatibility:	Not Compatible:
	Dimension uses regular expression for when to collect

Portlet State

Name:	Portlet State
Category:	WebSphere
Locked:	No
Dimension Type:	Standard
Based On:	Query Parameter: "PortletState"
When to Collect Data:	Hits that match the regex "PortletPID" is equal to ".*"
Multiple Values:	No
Drilldown Values:	No
Accumulation Type:	On Event
Visitor History Dependency:	Not Dependent
Express Analysis Compatibility:	Not Compatible:
	Dimension uses regular expression for when to collect

Portlet Drilldown

Name:	Portlet Drilldown
Category:	WebSphere
Locked:	No
Dimension Type:	Drilldown
Based On:	Custom Drilldown
	Portlet Mode
	Portlet State
When to Collect Data:	All Hits
Multiple Values:	No
Accumulation Type:	On Event
Visitor History Dependency:	Not Dependent
Express Analysis Compatibility:	Not Compatible:
	Dimension based on Custom Drilldown.
	Dimension contains drilldown data.

Command

Name:	Command
Category:	WebSphere
Locked:	No

Dimension Type:	Standard
Based On:	URL Regular Expression "/Command/(.*[^/])"
When to Collect Data:	All Hits
Multiple Values:	No
Drilldown Values:	No
Accumulation Type:	On Event
Visitor History Dependency:	Not Dependent
Express Analysis Compatibility:	Not Compatible: Dimension is based on a regular expression. Dimension uses regular expression for when to collect

Portlet Name

Name:	Portlet Name
Category:	WebSphere
Locked:	No
Dimension Type:	Standard
Based On:	URL Regular Expression " /Portlet/.*/(.*[^/])"
When to Collect Data:	All Hits
Multiple Values:	No
Drilldown Values:	No
Accumulation Type:	On Event
Visitor History Dependency:	Not Dependent
Express Analysis Compatibility:	Not Compatible: Dimension is based on a regular expression. Dimension uses regular expression for when to collect

User Management

Name:	User Management
Category:	WebSphere
Locked:	No
Dimension Type:	Standard
Based On:	URL Regular Expression " /Command/UserManagement/(.*[^/])"
When to Collect Data:	All Hits
Multiple Values:	No
Drilldown Values:	No
Accumulation Type:	On Event
Visitor History Dependency:	Not Dependent
Express Analysis Compatibility:	Not Compatible: Dimension is based on a regular expression. Dimension uses regular expression for when to collect

Page Customizer

Name:	Page Customizer
Category:	WebSphere

Locked:	No
Dimension Type:	Standard
Based On:	URL Regular Expression " .*Customizer\/(.*[^/])"
When to Collect Data:	All Hits
Multiple Values:	No
Drilldown Values:	No
Accumulation Type:	On Event
Visitor History Dependency:	Not Dependent
Express Analysis Compatibility:	Not Compatible: Dimension is based on a regular expression. Dimension uses regular expression for when to collect

IBM Pages

Name:	IBM Pages
Category:	WebSphere
Locked:	No
Dimension Type:	Standard
Based On:	URL with Query String
When to Collect Data:	
Accumulation Type:	On Event
Visitor History Dependency:	Not Dependent
Express Analysis Compatibility:	Not Compatible: Dimension based on URL with Query String.

Custom Report Filter

There is one custom filter that has been defined for use in the custom reports that are shown above. This filter is:

● Page filter

The sections that follow provide the setup information for this filter.

Page filter

Name:	Page filter
Category:	WebSphere
Locked:	No
Filter Type	Hit
Based On:	URLs that match the regex "/Page/.*"

Appendix A: The dcsMultiTrack() Function

When the dcsMultiTrack function is added to the Webtrends base JavaScript code, it provides the ability to begin tracking visitor behaviors that do not follow the standard page view strategy for tracking visitor behavior. The dcsMultiTrack function is a general function to track any visitor event that would not trigger an event.

Simply adding the dcsMultiTrack function will not automatically start tracking this data. For downloads and offsite links there are two techniques to track this site usage. Manually call the dcsMultiTrack for each link on your site or use the dcsDownload and dcsOffsite functions. These two techniques should not be mixed as it will result in two sets of the same data being sent to the Webtrends data servers for the same visitor event.

To track these events, the Webtrends Advanced JavaScript functions can be added to the existing Webtrends JavaScript implemented on your Web site. The following will describe what these functions do and how they can be utilized to send this additional data to the Webtrends Data Collection Servers.

Using the dcsMultiTrack() JavaScript function

The dcsMultiTrack function is part of the base Webtrends SmartSource JavaScript functions. The variables sent are in a comma-delimited format and are in a parameter name to value pairs. The following illustrates a basic example of a call to the dcsMultiTrack JavaScript function:

dcsMultiTrack('DCS.dcsuri','/custompage.aspx','WT.ti','The Page Title')

This call can be made as part of a mouse event, an event trigger, a link or can be called from a JavaScript function.

Any standard or custom parameter can be passed using the dcsMultiTrack function. However, there are two parameters that need to be passed when using dcsMultiTrack:

- DCS.dcsuri – the Page URI. This can be the actual page URI or a virtual page to uniquely identify the even the dcsMultiTrack is being used for.
- WT.ti – the page title to be displayed in the report.

For a complete list of Webtrends standard query parameters please see the Webtrends Query Parameters document in the Webtrends Customer Center available in the Webtrends UI.

Any additional parameters can be added by adding parameter name and value pairs to the list of parameters sent to the dcsMultiTrack function.

Common Technologies Which Use dcsMultiTrack()

HTML Anchor Tags – For HTML-based tracking, the dcsMultiTrack call is used in the anchor tag and bound to a mouse event, most commonly the OnClick event. A common example is tracking a pdf file:

 Corporate PDF

- Asynchronous JavaScript and XML (AJAX) – For AJAX implementations, the tagging strategy is to build a generic JavaScript function within the AJAX application to call the Webtrends dcsMultiTrack()

- Adobe Flash ActionScript – For Flash implementations, a generic ActionScript function and a few ActionScript libraries can be used to communicate with JavaScript in the .swf HTML wrapper.

What Happens When the dcsMultiTrack() Function is Used

When the dcsMultiTrack() function is used, it sends the defined parameter data to the Webtrends data collection servers. This generates a new hit with the defined parameters in the log files that is included in the reports. dcsMultiTrack should only be implemented for visitor behavior that is not normally tracked during a traditional page load.

The following is an example of data sent to Webtrends data collection servers after a traditional page load and then after clicking on a link using the HTML OnMouseDown event using dcsMultiTrack in the HTML sample:

```
<a href="CorporatePDF.pdf" OnMouseDown="dcsMultiTrack('DCS.dcsuri','/CorporatePDF.pdf','WT.ti','The Corporate PDF File');"> Corporate PDF </a>
```
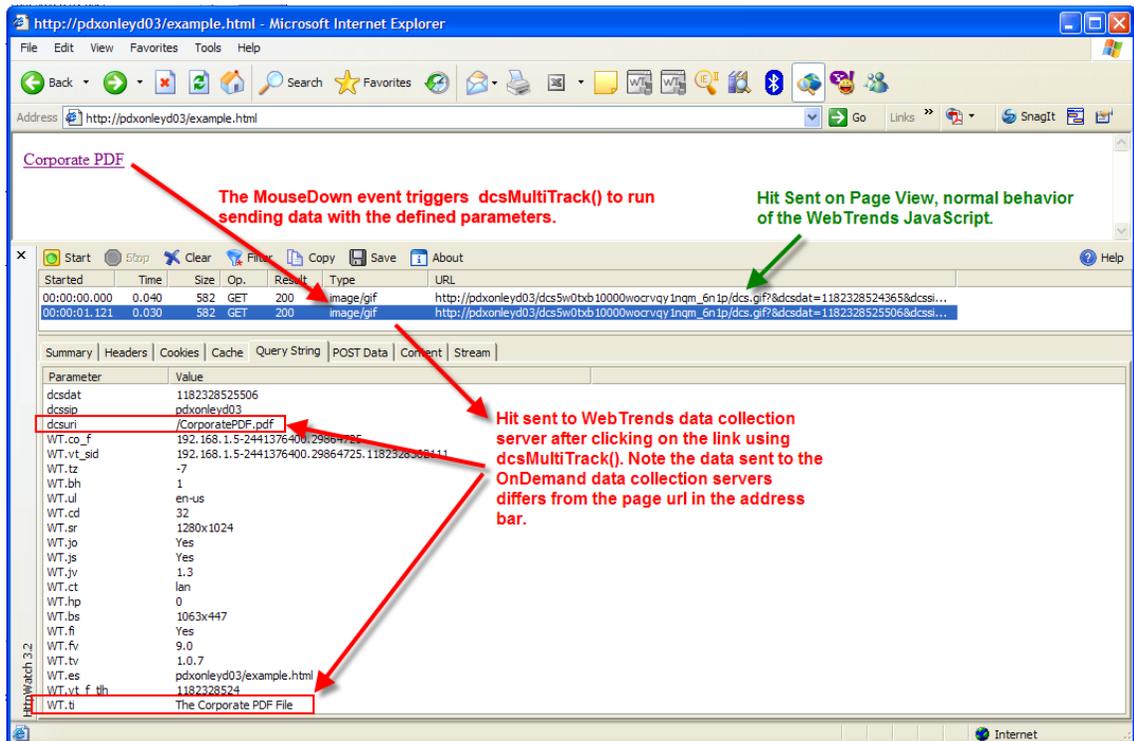

Note: Screenshot taken using HttpWatch.

The dcsDownload() and dcsOffsite() Advanced JavaScript Functions

dcsDownload() and dcsOffsite functions have been created and can be enabled to automatically send data when a link to a download (pdf, exe, etc) or an off-site link has been clicked. When appropriately implemented, these functions do not require additional tagging for these events which do not normally trigger data to be sent to the Webtrends data collection servers. These functions expect links to be delivered in a standard format.

DB2, DB2 Universal Database, Domino, IBM, the IBM logo, Everyplace, Informix, iNotes, Lotus, Lotus Notes, the On Demand Business logo, QuickPlace, Rational, Sametime, SmartSuite, Tivoli, WebSphere, Workplace and the Workplace family of marks, and z/OS are trademarks of International Business Machines Corporation in the United States, other countries or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries or both.

Microsoft, Windows and Windows NT are trademarks of Microsoft Corporation in the United States, other countries or both.

Other company, product and service names may be trademarks or service marks of others.

Contacts.

For questions about this or other Webtrends solutions, contact:

Michael Cook
Michael.cook@webtrends.com
(503) 553-2143

or

Abed Farhan
Abed.farhan@webtrends.com
(503) 553-2747

The information contained in this document represents the current view of Webtrends on the issues discussed as of the date of publication. Because Webtrends must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Webtrends, and Webtrends does not cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. WEBTRENDS MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Webtrends may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Webtrends, the furnishing of this document does not give you any license to these patents, trademarks, copyrights or other intellectual property.

Webtrends, Social Measurement, Ad Director, Analytics, Marketing Warehouse, Score, and Visitor Intelligence are trademarks of Webtrends Inc. All other company and product names may be trademarks of the respective companies with which they are associated.