QALoad 05.07

Using the Conductor

Technical support is available from our Customer Support Hotline or via our FrontLine Support Web site.

Customer Support Hotline:
1-800-538-7822

FrontLine Support Web Site:
http://frontline.compuware.com

Doc. CWQLHX
August 5, 2008

Table Of Contents

QALoad Online Help

Conductor

About the Conductor

Overview of the QALoad Conductor

Use the QALoad Conductor to configure, run, and monitor a load test that utilizes the scripts created in the Script Development Workbench. The Conductor controls the QALoad Players and manages tests while they are running. When the Conductor process stops for any reason during a load test, the associated Player processes automatically terminate.

The primary windows in the Conductor are:

- Conductor Start Page - This is the page that appears when you first open the Conductor. From the Start Page, you can open an existing Conductor session, create a new session, configure the Conductor, or open the Test Configuration Wizard.

Once you create or open a session, the main Conductor window appears. The Conductor's interface is dynamic — it changes depending whether you are setting up a test or running a test.

- Test Setup Interface - This is the first window that appears when you open a session in the Conductor. Before running a test, you must set up general test options, configure Player workstations, assign compiled test scripts to Players, and set up monitoring options. You then save the test setup in a file called a session ID. Once you have configured and saved a test session ID, you can reuse it without re-entering any test information.

  The Test Setup Interface presents two ways you can enter information about your test:

  - The Visual Designer is the Conductor's main window. The Visual Designer graphically displays a collection of icons and nodes that represent your test session. Use this to enter information about your test and set up the machines and scripts for the test.
  - The Grid View window presents another way to perform these functions. The Grid View is a dockable window that contains two tabs in which you can enter the information for your test session.

- Runtime Window - While a test is running, the Conductor interface changes to the Runtime Window, which facilitates monitoring of individual machines and Players, and displays real-time test results. You can view default graphs of performance data that are created for you by the Conductor and create custom graphs based on the data being collected during the test. You can save custom graph layouts in the session ID file and reuse them in future tests.

QALoad Conductor Menus and Toolbar Buttons

The Conductor's menus and toolbar buttons are dynamic; their content depends on whether you are opening the Conductor from the Start Page, preparing a test setup or running a test.

Start Page

File

Edit

Actions

Tools

Help

Start Page Toolbar Buttons

Test Setup

The Conductor Configuration and Setup Menus allow you to configure the Conductor and your specific test. The menus and toolbars are:

File

Edit

View

Actions

Tools

Help

Configuration and Setup Toolbar Buttons

Running a Test

Use the Conductor's Runtime menus and toolbar to control your running test and the data that is displayed at test time. The menu names are:

View

Runtime Windows

Graph

Actions

Tools

Runtime Toolbar Buttons

Accessing the Conductor

The following procedure describes how to start the Conductor.

To start the QALoad Conductor from Windows:

Click Start>Programs>Compuware>QALoad>Conductor. The Conductor Start Page appears.

To start the Conductor from the command prompt:

Type `conductor <session_file_name> /l /e /a /t`

The applicable parameters are defined in the following table.

Parameter	Definition
`/l` (Optional)	Creates a log file showing error messages and test status.
`/e` (Optional)	Exits the Conductor when the test completes.

/a (Optional)	Launches Analyze when the test completes.
/t (Optional)	Executes Conductor at a set time. Valid time formats are `/txx:xx` or `/txx/xx/xx /txx:xx`.

The Conductor start page appears.

Using the Conductor Start Page

When you first start the Conductor, the Start Page displays. This provides a concise selection of options to help you start defining or modifying a Conductor session.

From the Start Page, you can perform the following tasks:

Recent Session

In the Recent Sessions section, you can:

! Create a new test session

! Open an existing session

Tasks

In the Tasks section, you can:

! Open the Test Configuration Wizard

! Create monitoring tasks

! Edit monitoring tasks

! Discover and verify Player machines

! Configure the Conductor session options

Help

Use the selections int he Help section to access the QALoad online help and Compuware's FrontLine website.

Note: If you select the Don't show this panel again option, you'll skip the start page and go directly to the Conductor Session Startup dialog box when you open the Conductor. To display the Start Page again when you open the Conductor, you must open a session, go to Tools>Options>Startup, and select Show Start Page.

Test Setup Interface

Overview of the Test Setup Interface

The Conductor provides two methods for designing a test session: the Visual Designer and the Grid View.

Visual Designer

The Conductor's main window is the Visual Designer. The Visual Designer consists of three parts that you use to create the test session:

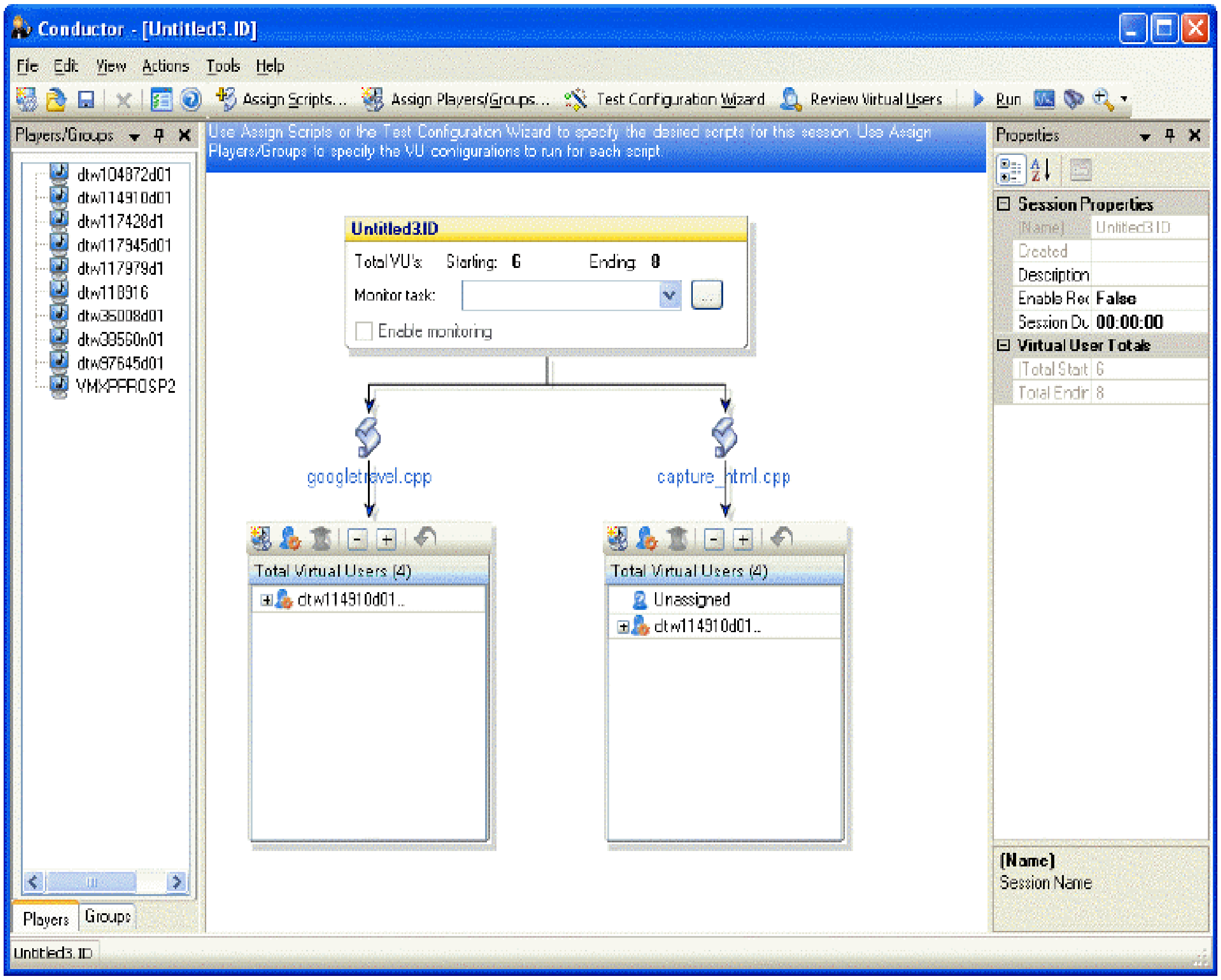
! Visual Designer window - contains a series of nodes displayed in a tree view. The session displays as the top-level node, while the scripts in the session are represented as nodes underneath the session.

! Players/Groups panel - This is a dockable panel the appears on the left-hand side of the window. It displays all Players and Groups available for the session.

! Properties panel - This is a dockable panel the appears on the right-hand side of the window. Select a session node, a script icon, or a Player machine in a script node to display information and set options for each one.

You can assign player machines to a script by dragging a player or a player group from the Player/Groups panel and dropping it into the script in the Visual Designer window. Review and update properties of the test elements in the Properties panel.

In addition, the Visual Designer's toolbar provides access to standard Windows functionality, such as Print and Copy, as well as quick access to Conductor setup options and to QALoad Analyze.



Grid View

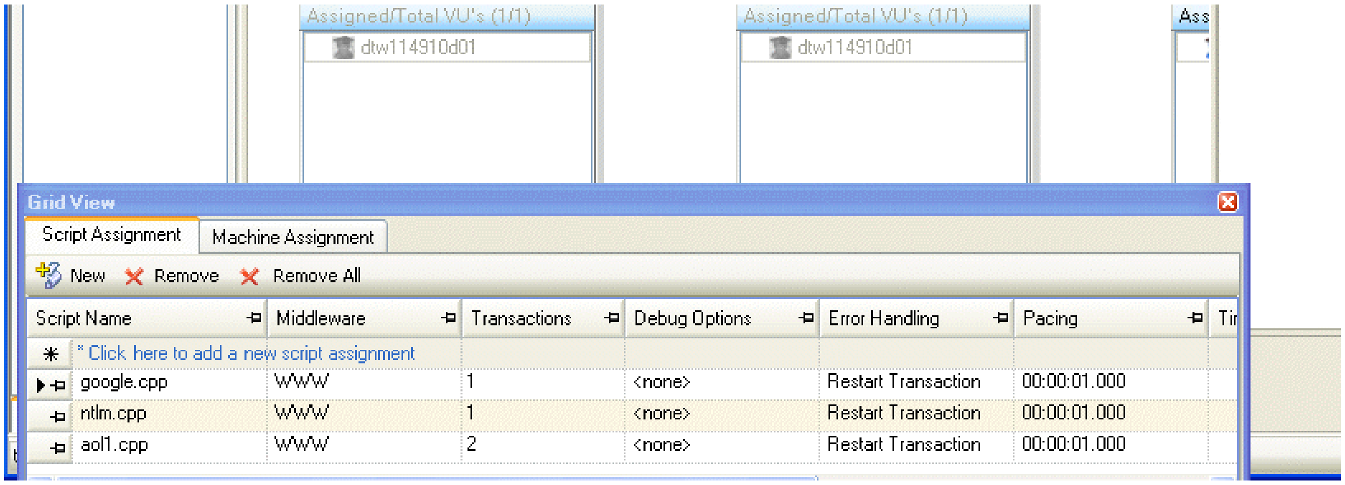
Alternately, you can open the Grid View to enter test information and set up the machines and scripts for the test. The Grid View is a dockable window that appears at the bottom of the Visual Designer window. Changes made in the Grid View appear in the Visual Designer.

The fields you use to set options for your load test are displayed in two tabs:

! Script Assignment tab - where you set up options for any scripts that have previously been recorded and compiled.

! Machine Assignment tab - where you assign scripts to specific Player workstations, starting and ending virtual users (VU), VU increments, and timing intervals.

The Grid View toolbars provides access to the main functions of assigning scripts and players to your load test session.



Script Name	Middleware	Transactions	Debug Options	Error Handling	Pacing
* Click here to add a new script assignment					
google.cpp	WWW	1	<none>	Restart Transaction	00:00:01.000
ntlm.cpp	WWW	1	<none>	Restart Transaction	00:00:01.000
aol1.cpp	WWW	2	<none>	Restart Transaction	00:00:01.000

Visual Designer

Using the Visual Designer

The Visual Designer is the main window, the test setup interface, that displays a collection of icons and nodes that represent your test session. The top level node is the current test session, and the child nodes represent the scripts assigned to the session.

Use the Visual Designer to enter information about your test and set up the machines and scripts for the test using the two panels in the Visual Designer window:

! Players/Groups panel - This is a two-tabbed window that list the installed QALoad Players and any groups in which they are members. Assign Players and groups to individual scripts by dragging and dropping them into the script nodes.

! Properties panel - This is a dynamic panel on the right-hand side of the Visual Designer that changes content depending on the view you choose. Use the Properties panel to setup and review configurations for session, script, and Player properties.

o Session properties - Click the top-level Session node to display session properties in the Session Properties panel. Use the Session Properties panel to enable recording and set session duration.

- Script properties - Click the script icon for any individual script to display its properties in the Script Properties panel. Use the Script Properties panel to set options for Application Vantage, external files and datapools, and additional script properties, such as debug options, error handling, number of transactions, and timing options.
- Player properties - Click an individual Player in a script node to view it's properties in the Player Properties panel. You can set starting and ending virtual users (VU), VU increments, mode and timing intervals, and set expert user options.

Use the toolbar buttons to assign scripts, assign players and groups, or to open the Test Configuration Wizard, where you can easily configure your test session.

Using the Properties Panel

Use the Visual Designer's Properties panel to review test information and setup test options for:

Sessions - The session properties are displayed when you click on the Session node of the Visual Designer.

Scripts - Script properties for an individual script are displayed when you click on the script icon for the selected script in the Visual Designer.

Players - Player properties for an individual Player are displayed when you click on a selected Player within the script node in the Visual Designer.

Using the Player/Groups Panel

The Players/Groups panel is a dockable, tabbed window on the left-hand side of the Visual Designer window. It display a list of all players and all groups available for this session. You can drag and drop individual Players and Player groups to the script node in the Visual Designer to assign the script to the Player or group.

To display available Players:

Click the Players tab at the bottom of the Players/Groups window to display a list of all players available for this session.

To display available Player groups:

Click the Groups tab at the bottom of the Player/Groups window to display a list of all groups to which available players belong.

Grid View

Using the Grid View

You can use the Grid View to enter test information about your test and set up the machines and scripts for the test. The Grid View contains two tabs:

- Script Assignment tab - Use this tab to set up options for any scripts that have previously been recorded and compiled. Any scripts you add here are included in your load test, and one virtual user is automatically assigned to your script on the Machine Assignment tab. After setting up your scripts here, you must assign additional virtual users to your script from the Machine Assignment tab.

- Machine Assignment tab - Use the Machine Assignment tab in the Grid View to assign scripts to specific Player workstations. You also assign starting and ending virtual users (VU), VU increments, and timing intervals.

To open the Grid View:

Click View>Grid Window. The Grid View pane appears below the Visual Designer window.

Using the Script Assignment Tab

Use the Script Assignment tab in the Grid View window to assign scripts to a test session, set middleware options for SAP and Citrix scripts, assign external data files to a script, and set general script options. Use the Script Assignment toolbar to add or remove items. Options you select in the Grid View window are reflected in the Script Properties panel on the right-hand side of the Visual Designer window.

To access the Script Assignment tab:

1. In the Conductor, click View>Grid Window. The Grid View window appears at the bottom of the Visual Designer window.
2. Click the Script Assignment tab.

Use the Script Assignment tab to:

Assign a script to a test session

Assign External Data options

You also can set:

Middleware Options for SAPGUI and Citrix

Number of transactions

Debug Options

Error Handling Options

Pacing

Timing options

Sleep Factor percent

Using the Machine Assignment Tab

Use the Machine Assignment tab in the Grid View window to replace scripts in a test session, assign a player to a script, set Expert User options, and set Virtual User configurations. Use the Machine Assignment toolbar to add, remove, or sort items in the display. Options you select in the Grid View window are reflected in the Player Properties panel on the right-hand side of the Visual Designer window.

To access the Machine Assignment tab:

1. In the Conductor, click View>Grid Window. The Grid View window appears at the bottom of the Visual Designer window.
2. Click the Machine Assignment tab.

Use the Machine Assignment tab to:

Replace a script in a test session

Specify Virtual User Configurations

Enable the Expert User (WWW only)

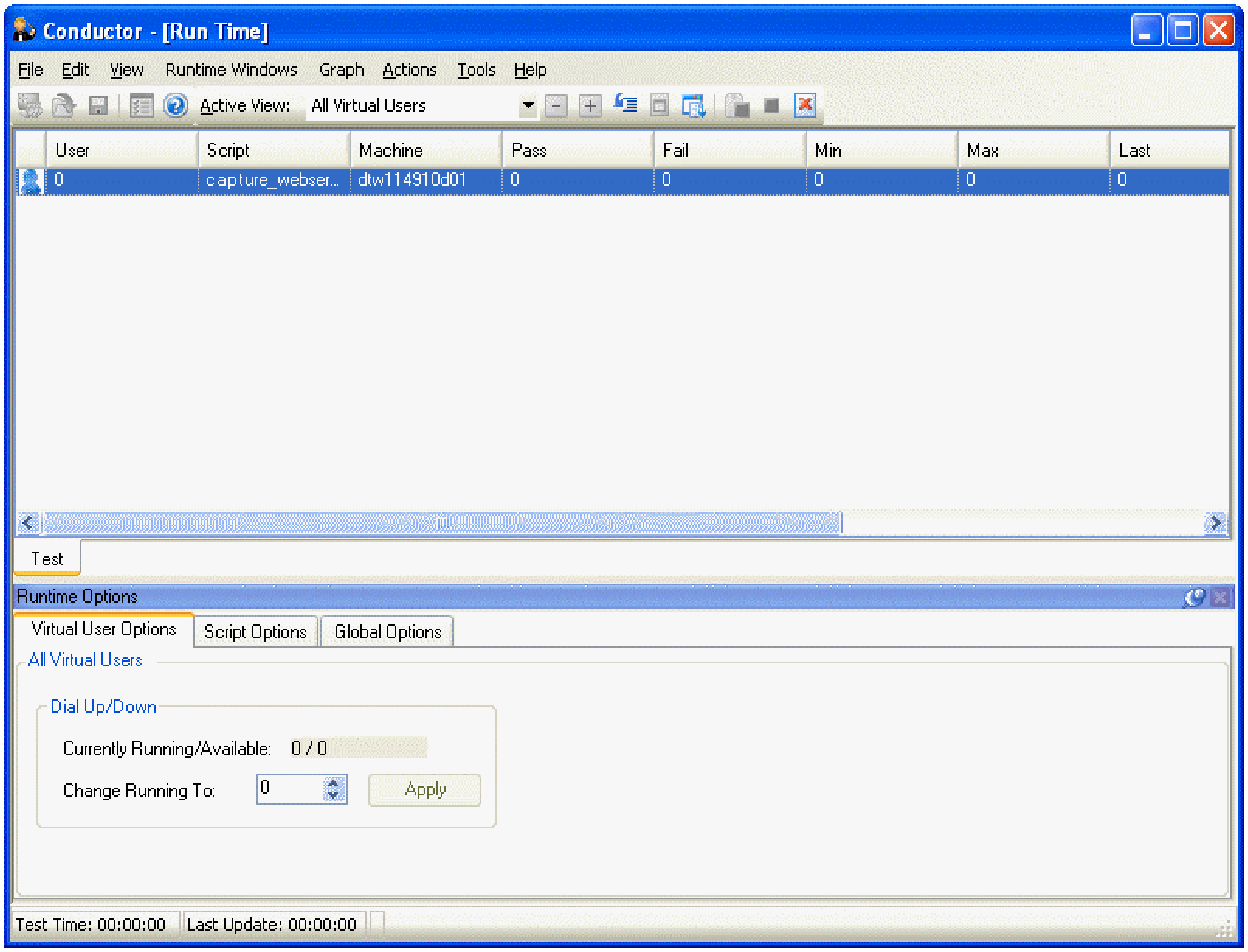
Runtime Window Interface

Runtime Window Interface

When you start a test, the Conductor's interface changes to an interactive test control station called the Runtime Window. The Runtime Window displays information about the scripts, machines, and virtual users that are executing the load test. The Runtime window is divided into two areas – the main Runtime window, and the dockable Runtime Options panel at the bottom of the window.

On the Runtime Window, you can observe the progress of individual scripts and Player machines, view real-time graphs, and start or suspend scripts and Players from a running test to better simulate the unpredictability of real users. The Runtime main window dynamically displays test details according to the type of information you select in the Active View field.

The lower pane, the Runtime Options panel, displays data for the current view and the individual script you select. This is a dockable control station that allows you to change virtual user options and data transfer options while your test is running.



Active View: All Virtual Users

User	Script	Machine	Pass	Fail	Min	Max	Last
0	capture_webser...	dtw114910d01	0	0	0	0	0

All Virtual Users

Dial Up/Down

Currently Running/Available: 0 / 0

Change Running To: 0

Test Time: 00:00:00 Last Update: 00:00:00

Runtime Window

When you start a test, the Conductor's interface changes to an interactive test control station called the Runtime Window. From the Runtime Window, you can observe the progress of individual scripts and Player machines, create real-time graphs, and change the behavior of scripts and Players from a running test to better simulate the unpredictability of real users. This window has two unique areas:

- Runtime main window - The information in the main Runtime window depends on the view you select in the Active View field. You can view details for all test scripts, individual test scripts, all player machines, and individual player machines.
- Runtime Options Panel - The lower pane, called the Runtime Options Panel, displays data for the current view and the individual script you select. This is a dockable control station that allows you to change virtual user options and data transfer options while your test is running. The information in the Runtime Options panel is displayed in three tabs: Virtual User Options Tab, Script Options Tab, Global Options Tab.

Elements of the Runtime Window

Active View Details

Details for a running load test are displayed in the Runtime main window. The details displayed depend on the option you select in the Active View field. You can choose:

All Virtual Users

Virtual Users by Script

Virtual Users by Machine

All Scripts

All Player Machines

Graphs View

Sessions View

All Virtual Users

Lists the script and player machine involved in the test and other detail information by each virtual user. Double-click a virtual user to display the Virtual User Info window. Select this option to view details about each virtual user running a script in the test:

Status icon: The first column displays an icon that indicates the status of the virtual user. A moving icon represents a running virtual user; a still icon represents a virtual user that hasn't yet started or is suspended; and an icon with a checkmark through it represents a virtual user that has exited. A red circle with an X indicates errors have occurred on that virtual user, or that the test session was manually terminated using the Exit, Abort, or Quit Current Test buttons.

User: Virtual User's identification number assigned by QALoad.

Script: Name of the script the virtual user is running.

Machine: Name of the Player machine on which the script is running.

Pass: Number of transactions successfully completed.

Fail: Number of transaction that have failed to complete successfully.

Min: Lists the shortest response time recorded for a transaction on the virtual user.

Max: Lists the longest response time recorded for a transaction on the virtual user.

Last: Lists the most recent response time recorded on the virtual user.

Status: Lists a status for the virtual user if any errors have been encountered, or lists the name of the checkpoint the user has encountered if the option Send all timing data including Checkpoint information was selected on the Runtime window of the Session Options dialog box during test setup. Double-click a virtual user for more information, or if the error message is to long to read.

Errors:

Error Message:

Note: Double-click in any field to display the Virtual User Info dialog box with the details about each virtual user.

Virtual Users by Script

Lists Virtual User and Player machine information within each assigned script. Double-click a virtual user to display the Virtual User Info window. Select this option to view the following details for each virtual user sorted by script:

User: Virtual User's identification number assigned by QALoad.

Machine: Name of the Player machine on which the script is running.

Pass: Number of transactions successfully completed.

Fail: Number of transaction that have failed to complete successfully.

Min: Lists the shortest response time recorded for a transaction on the virtual user.

Max: Lists the longest response time recorded for a transaction on the virtual user.

Last: Lists the most recent response time recorded on the virtual user.

Status: Lists a status for the virtual user if any errors have been encountered, or lists the name of the checkpoint the user has encountered if the option Send all timing data including Checkpoint information was selected on the Runtime window of the Session Options dialog box during test setup. Double-click a virtual user for more information, or if the error message is to long to read.

Errors:

Error Message:

Note: Double-click in any field to display the Virtual User Info dialog box with the details about each virtual user.

Virtual Users by Machine

Lists Virtual User and script information within each assigned player machine. Double-click a virtual user to display the Virtual User Info window. Select this option to view the following details for each virtual user sorted by Player machine:

User: Virtual User's identification number assigned by QALoad.

Script: Name of the script the virtual user is running.

Pass: Number of transactions successfully completed.

Fail: Number of transaction that have failed to complete successfully.

Min: Lists the shortest response time recorded for a transaction on the virtual user.

Max: Lists the longest response time recorded for a transaction on the virtual user.

Last: Lists the most recent response time recorded on the virtual user.

Status: Lists a status for the virtual user if any errors have been encountered, or lists the name of the checkpoint the user has encountered if the option Send all timing data including Checkpoint information was selected on the Runtime window of the Session Options dialog box during test setup. Double-click a virtual user for more information, or if the error message is to long to read.

Errors:

Error Message:

> Note: Double-click in any field to display the Virtual User Info dialog box with the details about each virtual user.

All Scripts

Lists detail information for each script assigned to the test. Select this option to display the following summary details about every script in the test:

Script: The name of the script.

Total Users: The total number of virtual users assigned to run the script.

Running: The total number of virtual users currently running the script. This number may vary at different times in a single test if you are configured for dial-up virtual users, or have configured the test as a ramp-up test.

Pass: The number of transactions successfully completed.

Fail: The number of transactions that have failed to complete successfully.

Response Time: The average response time of all virtual users currently running the test script.

Throughput: The average number of transactions per second this script is running.

All Player machines

Lists detail information for each player machine assigned to the test. Select this option to display the following summary information about each Player machine running the test.

Machine: The machine name, preceded by an icon indicating whether the test is currently running on the machine or finished. A blue checkmark indicates a successfully completed test.

Total Users: The total number of virtual users controlled by that Player Agent.

Running: The total number of virtual users currently running on that Player Agent machine.

% Processor: The percentage of the Player Agent machine's processor that is currently in use.

% Memory: The percentage of the Player Agent machine's memory that is currently in use.

% Disk: The percentage of the Player Agent machine's disk space that is currently in use.

Status: Lists a general status for the test. For example, Test is running.

Graphs View

Displays real-time graphs for checkpoints, performance counters, and Player machine health statistics. You can control which types of data are graphed in addition to how the graphs appear. For detailed information, refer to Graphs View.

Sessions View

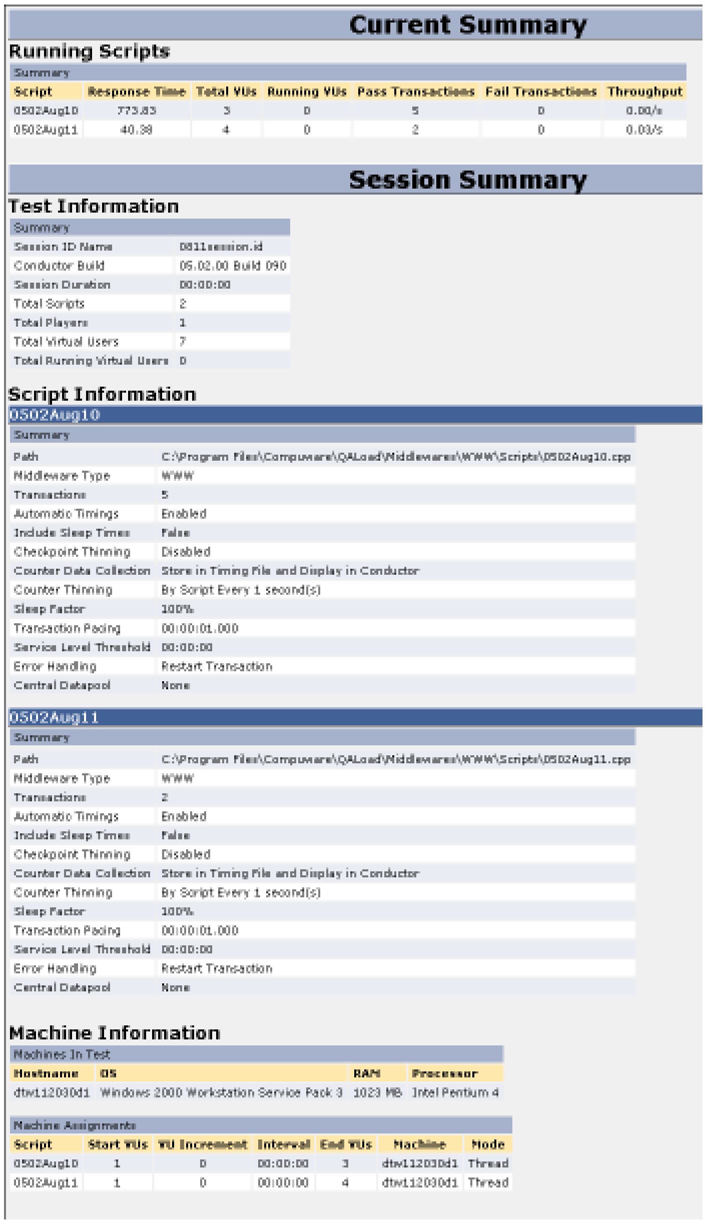
Displays summary information about the test and the Player machine. For more information, refer to Session View

Session View

When you select Session in the Active View field, the Conductor Runtime Window provides summary information about the test session that is currently running. The Session view can be printed as a report by right-clicking and choosing Print from the shortcut menu.

Note: The Session view below has been cropped to better fit this help topic, while still representing what a real Session view might look like.

Click on the sections in the following graphic for more information about the Session view.



Current Summary

Running Scripts

Summary

Script	Response Time	Total VUs	Running VUs	Pass Transactions	Fail Transactions	Throughput
0502Aug10	773.83	3	0	5	0	0.00/s
0502Aug11	40.38	4	0	2	0	0.03/s

Session Summary

Test Information

Summary	
Session ID Name	0811session.id
Conductor Build	05.02.00 Build 090
Session Duration	00:00:00
Total Scripts	2
Total Players	1
Total Virtual Users	7
Total Running Virtual Users	0

Script Information

0502Aug10

Summary	
Path	C:\Program Files\Compuware\QALoad\Middlewares\WWW\Scripts\0502Aug10.cpp
Middleware Type	WWW
Transactions	5
Automatic Timings	Enabled
Include Sleep Times	False
Checkpoint Thinning	Disabled
Counter Data Collection	Store in Timing File and Display in Conductor
Counter Thinning	By Script Every 1 second(s)
Sleep Factor	100%
Transaction Pacing	00:00:01.000
Service Level Threshold	00:00:00
Error Handling	Restart Transaction
Central Datapool	None

0502Aug11

Summary	
Path	C:\Program Files\Compuware\QALoad\Middlewares\WWW\Scripts\0502Aug11.cpp
Middleware Type	WWW
Transactions	2
Automatic Timings	Enabled
Include Sleep Times	False
Checkpoint Thinning	Disabled
Counter Data Collection	Store in Timing File and Display in Conductor
Counter Thinning	By Script Every 1 second(s)
Sleep Factor	100%
Transaction Pacing	00:00:01.000
Service Level Threshold	00:00:00
Error Handling	Restart Transaction
Central Datapool	None

Machine Information

Machines In Test

Hostname	OS	RAM	Processor
dtw112030d1	Windows 2000 Workstation Service Pack 3	1023 MB	Intel Pentium 4

Machine Assignments

Script	Start VUs	VU Increment	Interval	End VUs	Machine	Mode
0502Aug10	1	0	00:00:00	3	dtw112030d1	Thread
0502Aug11	1	0	00:00:00	4	dtw112030d1	Thread

Graphs View

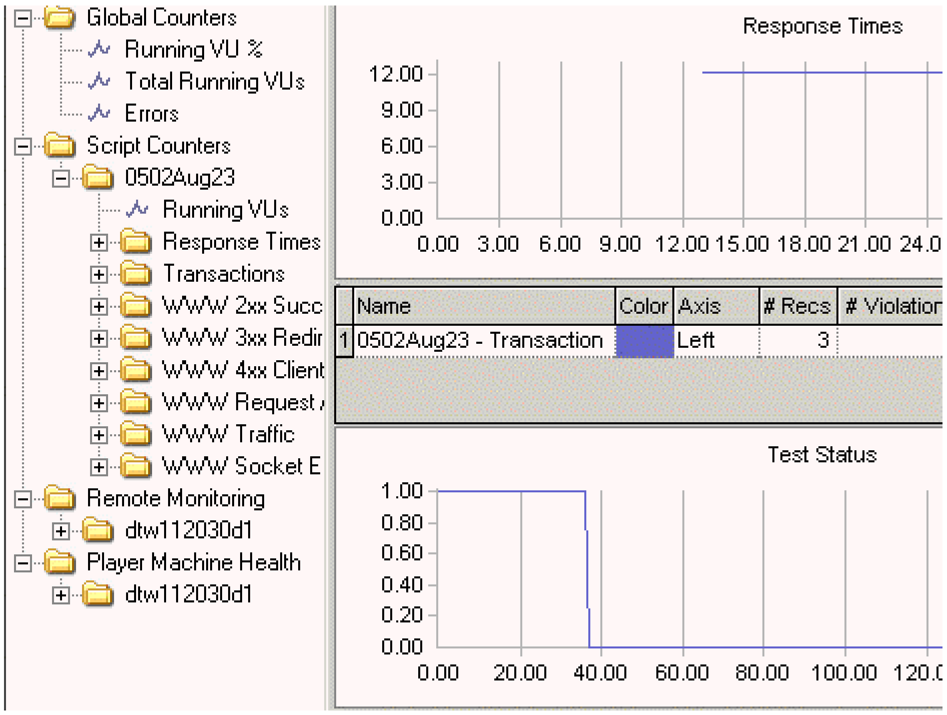
The Graphs view in the Conductor Runtime Window displays graphs of data collected during the test. By default, the Graphs view displays graphs for response times, test status, and player machine health.

Other graphs, such as user-defined checkpoints and Remote Monitoring counters, can also be plotted in the right pane of the Graphs view if they were enabled for the session.

To display graphs:

1. Right-click on a counter or other data type in the tree view that you want to plot in a graph.
2. Choose Add Graph or Add Plot To.

You can also modify a graph's appearance by right-clicking on the graph and choosing one of the formatting options, such as colors and axes properties. To increase the visibility of a plot when you have multiple plots on a graph, click on a plot (or that plot's number in the legend) to highlight it.



	Name	Color	Axis	# Recs	# Violation
1	0502Aug23 - Transaction	🟦	Left	3	

Runtime Options Panel

This dockable control station enables you to change virtual user options and data transfer options while your test is running. The information in the Runtime Options panel is displayed in three tabs:

Virtual User Options Tab

Currently Running/Available: Displays the number of currently running Virtual Users.

Change Running To: Click the arrows to change the number of Virtual Users in the test.

Apply: Click to apply your changes.

Script Options Tab

Changes made in this window will override options set during the original Script Assignment.

Note: Options on this tab are available only when Virtual Users by Script or All Scripts are selected in the Active View.

Error Handling: Choose how to respond when an error occurs during execution of the transaction. During large load tests, errors can sometimes indicate that the test is straining the limits of the hardware/software in the testing environment. Options are:

- Abort Transaction — If an error occurs while a transaction is being executed, the Player should abort the current transaction and the virtual user who encountered the error should exit the test. Use this option when errors will make the virtual user invalid for executing more transactions.
- Continue Transaction — If an error occurs while a transaction is being executed, the Player should continue executing the transaction as if the error didn't occur. Select this option when errors are not critical to the performance of the load test and can be safely ignored.
- Restart Transaction (WWW, SAPGUI, and Citrix scripts only) — If an error occurs while a transaction is being executed, the Player should abort the current transaction entirely and restart a new transaction from the beginning. Note that the transaction count will increase for each transaction that is restarted.

Pacing: Enter a value in this field to change the rate of pacing. Pacing is the time interval between the start of a transaction and the beginning of the next transaction on each workstation running the script. For example: if a transaction is designed to duplicate the process of someone handling incoming telephone calls and those calls arrive at a rate of 40 per hour/per person, set the pacing rate at 90 seconds.

Sleep: QALoad records the actual delays between requests and inserts the DO_SLEEP command in the script to mimic those delays when the script is played back in a test. You can maintain the exact length of the recorded delays at playback, or shorten them by entering a smaller percentage of the originally recorded delay to play back. For example, if you recorded a delay of 10 seconds then DO_SLEEP (10); is written to your script. Then, if a Sleep Factor of 50% is specified here, the Player will sleep for 5 seconds at that statement when the test is executed.

Apply: Click to apply your changes to the running script.

Cancel: Click to cancel any changes you have not yet applied to the running script.

Global Options Tab

Global Options apply to all scripts.

Timing Updates: Select when the Players in your test should send timing information to the Conductor. You can choose:

No Updates - No timing updates are sent to the Conductor

Send All - Sends all timing updates to the Conductor

Periodic Updates - If you chose Periodic Updates, type how often, in seconds, timing updates should be sent to the Conductor in the (1 - 1000 Sec.) field.

Apply: Click to apply your changes to the running script.

Cancel: Click to cancel any changes you have not yet applied to the running script.

Setting Up the Conductor

About Setting Up a Test

When you set up a load test, you set options related to general Conductor behavior as well as information about your specific test environment. Before you can successfully set up a load test, you must have recorded and compiled one or more test scripts. For information about recording a test script, see Developing Scripts.

Determining General Conductor Behavior

General Conductor options you set are applicable for all tests run until you change the options. Conductor options are related to the following:

- Viewing options for real-time results
- Global Player options
- Player machine performance data
- Options for runtime reporting
- And more...

All of the above information, and more, can be configured on the Conductor's Options dialog box.

Setting Up a Specific Test Session

To prepare the Conductor for a specific test, save information and parameters specific to that test into a reusable session ID file (.id). You must enter the following types of information to set up a test's session ID file:

- General information about the test such as a description, the size of the database, the length of the test, and any notes or comments
- Information about the test script(s) included in the test, including script name, middleware/protocol type, pacing, whether to include external data, and so on
- Information about the workstations where the QALoad Players reside, including which script is assigned to each workstation, how many virtual users are assigned to each workstation, the machine name, and so on.

All of the above information can be entered and saved when you set up a test from the Conductor's main window or by using the Test Configuration Wizard. Once you open the new session in the Conductor, you can do the following:

- (Optional) configuration for server monitoring
- (Optional) integration with other Compuware products

Generating Random Number Seeds

Random number seeds are used to inject random delays in script execution for each load test. The seed (or value) is automatically generated by QALoad. The random value used within the end of transaction function is used to generate the pacing time. The Player uses a system-generated sequence of numbers, so that each virtual user (VU) has its own seed value.

Setting Up the Conductor

To prepare for running a load test, you must set up the Conductor.

To set up the Conductor:

1. Start the Conductor.
2. Configure the Conductor. After starting the Conductor, you may need to verify that the Conductor’s configuration parameters are set properly.
3. Set Up a Session ID File. For every test you run, you must create a session ID file containing information the Conductor needs to run the test, such as which scripts to run, which Player machines to use, and whether to collect server or performance monitoring data. Use the Conductor's Visual Designer or Grid View window to create and save session ID files in the \QAload\Session directory.

Configuring the Conductor

There are several settings for the Conductor that you should review before beginning your load test.

To set Conductor session options that are not specific to one test:

1. Do one of the following:

   ! From the Conductor Start Page, click Session Options in the Tasks area.

   OR

   ! With a session open, click Tools>Options.

2. On the Options dialog box, set options related to post-test activity, warnings and prompts, runtime grids, timing settings, interface refresh intervals, Conductor/Player communications, monitoring intervals, and more. For detailed descriptions of the options that are available, see Options dialog box.
3. When you are finished, click OK to save your changes. Any options you set apply to all tests until you change them.

Setting Up a Test

Starting a Session

Creating a New Session

When you open the Conductor, the Conductor Start Page appears. Use these procedures to create a new session.

Note: If you choose not to display this page, the Conductor opens to the Visual Designer window. You can choose to show the Start Page again in the Startup page of the Conductor Sessions Options dialog box.

To create a new session from the Conductor Start page:

1. Do one of the following:

   ! In the Recent Sessions area, click New.

   OR

   ! In the Tasks area, click New Session.

2. In the Name field of the Create New Session dialog box, type a name for the test session.
3. (Optional) In the Description field, type a description for the session.
4. (Optional) Select Launch Test Configuration Wizard to open the Test Configuration Wizard to create your session.
5. Click OK.

If you do not select the option to use the Test Configuration Wizard, the Conductor opens with the session you created displayed in the Visual Designer. You can begin to setup your test session.

To create a new session in the Visual Designer:

1. With a session open in the Visual Designer window, click File>New. A confirmation dialog box prompts you to Save or discard changes to your current session. The Create New Session dialog box appears.
2. In the Name field, type a name for the new session.
3. (Optional) In the Description field, type a description for the new session.
4. (Optional) Select Launch Test Configuration Wizard to start the Test Configuration Wizard, which walks you through creating a load test session. If you do not select this option, the new test session opens in Conductor's Visual Designer, where you can manually configure the load test session.

Opening an Existing Session

When you open the Conductor, the Conductor Start Page appears. Use these procedures to open an existing session.

Note: If you choose not to display this page, the Conductor opens to the Visual Designer window. You can choose to show the Start Page again in the Startup page of the Conductor Sessions Options dialog box.

To create a new session from the Conductor Start page:

1. Do one of the following:
   - In the Recent Sessions area, click the name of the session to open.

   OR

   - Click Open Session, then select the name of the appropriate saved session, and click Open.
2. The Conductor's Visual Designer opens with the session you selected displayed. You can make any changes to the session or run the load test.

To open an existing session in the Visual Designer:

1. From the Visual Designer window, click File>Open. A confirmation dialog box prompts you to Save or discard changes to your current session.
2. Select the name of the appropriate saved session, and click Open. The selected test session opens in Conductor's Visual Designer, where you can manually configure the load test session.

Setting Up a Test Session

You can enter all the information necessary for your session ID file in the Conductor using one of the following methods:

- Grid View window
- Visual Designer
- Test Configuration Wizard

Assigning Scripts to the Test Session

Script Assignment

Use the Assign Scripts button on the Visual Designer toolbar, or the Script Assignment tab in the Grid View window to set up any scripts that have previously been recorded and compiled. Any script you add here is included in your load test, and one virtual user is automatically assigned to your script. After setting up your scripts here, you must assign additional virtual users to your script from the Properties pane in the Visual Designer, or from the Machine Assignment tab in the Grid View window.

You can also add scripts to a test session using the Test Configuration Wizard. For more information, see About the Test Configuration Wizard.

Adding a Script to a Test

To add a script to a test session:

1. On the Visual Designer toolbar, click Assign Scripts...

   OR

   In the Grid View window, click the Script Assignment tab, then click New in the toolbar. The Assign Scripts dialog box displays.
2. In the Middleware Type box, select your middleware type.
3. From the list of available scripts that appears in the Available Scripts pane, highlight a script name and click ➡ . The script is moved to the Selected Scripts pane.

Note: To select more than one script, hold down the Ctrl key and click then scripts to select, the click ➡. To select all scripts in the Available Scripts pane, click ⇥.

4. Click Next to display the script transaction configuration dialog box.
5. In the Selected Scripts pane, highlight a script.
6. In the Transactions field, specify the maximum number of transactions that you want each virtual user running this script to run. Once a workstation executes the number you specify, script execution continues with the line following the End_Transaction command rather than jumping to the beginning of the transaction loop.
7. Enter a value, in seconds, in the Pacing field. Pacing is the time interval between the start of a transaction and the start of the next transaction for each virtual user running a script.
8. Enter a value in the Sleep Factor % field to specify the percentage of any originally-recorded delay to preserve in the script (for example, a value of 80 means preserve 80% of the original delay). Valid values are 0-1000, or Random. The default value is 100%.
9. To apply these options to all scripts in the Selected Scripts pane, click Apply this to all selected scripts.
10. Click Finish to save your changes to the current session ID file.

Replacing a Script in a Test

You can replace a script in a test session using the Visual Designer or the Grid View window. Use the following procedure to replace a script in a test:

To replace a script using the Visual Designer:

1. In the Conductor's menu bar, click Actions>Replace Script. The Select Replacement Script dialog box appears.
2. In the Middleware Type field, select the middleware environment of the replacement script.
3. Select the replacement script, then click OK.
4. Assign the script to Player machines, if necessary.

To replace a script using the Grid View window:

1. In the Grid View window, click the Machine Assignment tab.
2. In the Script Name field, select the script to replace, then select the new script from the drop-down dialog box.
3. Assign the script to Player machines, if necessary.

Removing a Script or a Player from a Test

To remove a script from a test:

1. Right-click on the script icon for the appropriate script, then select Remove Script.
2. In the confirmation dialog box, click Remove.

To remove all scripts from a test:

1. Click Actions>Remove All Scripts.
2. In the confirmation dialog box, click Remove.

To remove a single Player from a test:

1. In the script test setup node, right-click on the Player to remove, then select Remove Selected.
2. In the confirmation dialog box, click Remove.

To remove all Players from a test:

1. Click Actions>Remove All Players/Groups.
2. In the confirmation dialog box, click Remove.

Assigning Player Machines

Machine Assignment

Use the Assign Players/Groups button on the Visual Designer toolbar, or use the Machine Assignment tab in the Grid View window to open the Assign Players/Groups dialog box and assign scripts to specific Player workstations. You also can drag and drop individual Player machines in selected script test setup nodes in the Visual Designer.

Assigning Scripts to Player Workstations

Once you've added scripts to your test session, you must assign scripts to Player workstations and set up Virtual User configurations. You can use:

! the drag and drop method from the Players/Groups panel

! the toolbar in the Visual Designer

! the Machine Assignment tab in the Grid View

Note: You cannot run multiple OFS scripts with different Forms Environment settings or different Connection Mode settings on the same player.

To assign players using the drag and drop method from the Players/ Groups panel:

Note: Use the Player Properties panel in the Visual Designer to set the optional Expert User options for WWW scripts.

1. Select a machine or group from the list in the Players/Groups window, then drag and drop it into the appropriate script test setup node. The Configure All Players/Groups dialog box appears.
2. Use the arrow keys in each field to set the following options:

   ! Starting VUs - Number of Virtual Users to launch the script on the machine when the test begins.

   ! Ending VUs - Number of Virtual Users at the end of the test.

   ! VU increment - Number of virtual users to add at intervals throughout the test.

   Note: If you add incremental virtual users, you must designate the time interval and the ending virtual users.

   ! Time interval - Time interval at which incremental virtual users should be added to the test.

   ! Mode - The test mode for the machine. You can select thread based or process based.
3. To use these options for all players and groups in the script, click Assign to all selected players and groups.
4. Click Finish.
5. Click File>Save in the Conductor main menu.

To assign players using the toolbar in the Visual Designer:

Note: Use the Player Properties panel in the Visual Designer to set the optional Expert User options for WWW scripts.

1. Click the appropriate script icon in the Visual Designer.
2. Click the Assign Players/Groups button in the toolbar. The Assign Players/Groups dialog box appears.
3. Select the Player machine or group, then click Next to display the Configure All Players/Groups dialog box.
4. Click the arrows in the fields to set the following options:

   ! Starting VUs - Number of Virtual Users to launch the script on the machine when the test begins.

   ! Ending VUs - Number of Virtual Users at the end of the test.

- ! VU increment - Number of virtual users to add at intervals throughout the test.

Note: If you add incremental virtual users, you must designate the time interval and the ending virtual users.

- ! Time interval - Time interval at which incremental virtual users should be added to the test.
- ! Mode - The test mode for the machine. You can select thread based or process based.

5. To use these options for all players and groups in the script, click Assign to all selected players and groups.
6. Click Finish.
7. Click File>Save in the Conductor main menu.

To assign players using the Grid View:

1. In the Machine Assignment tab, click the down arrow in the Player Name field, then select a player from the list.
2. (Optional - WWW only) In the Middleware field, click the browse [...] button to open the Expert User Options dialog box. If you enable the Expert User, select the Virtual User number to represent the Expert User.
3. Click each of the following fields to set options for:
   - ! Starting VUs - Type the number of Virtual Users to launch the script on the machine when the test begins.
   - ! Ending VUs - Type the number of Virtual Users at the end of the test.
   - ! VU Increment - Use the arrows in the field to set the number of virtual users to add at intervals throughout the test.
   - ! Timing Interval - Click the browse [...] button in the field to open the Set Time Interval dialog box. Set the time interval at which incremental virtual users should be added to the test.

   Note: If you add incremental virtual users, you must designate the time interval and the ending virtual users.

   - ! Mode - Use the arrows in the field to select the test mode for the machine. You can select thread based or process based.
4. Click File>Save in the Conductor main menu.

Adding Player Machines to a Test Session

Follow these instructions to add a Player workstation to your pool of available Players in a test's session ID file.

To add a new Player machine:

1. From the Conductor's main menu, click Tools>Manage Players. The Manage Players/Groups dialog box displays.
2. Click File>New>Player. A new page displays in the detail area of the dialog box, where you enter information and settings for the new Player.

To enter information and settings for the new Player:

Enter information for the new Player in the following areas of the dialog box.

In the Player Information area:

1. In the Machine (hostname or IP) field, type a name for the Player Machine.
2. In the Communications [TCP] port field, type the port number the Conductor should use to communicate (using TCP) with this machine during a test. The default is 3032.
3. Click Verify to check that the machine is active. The Player or Agent returns the operating system, processor type, amount of memory, and the maximum threads and processes available on the machine.

Note: If a Player does not respond, a message box appears indicating that the Player is not responding. If the Player is not responding, one of the following scenarios is likely:

- The host name and/or port number you entered may not be correct. Check your parameters and network connections, then try to send another request.
- The Player is not running. Start the Player and then try to send another request.

Expand the Machine Settings area (Optional):

4. Select desired options to ping the host before attempting connection to the player, generate IP spoof data, or override the default machine settings.
5. Close the Machine Settings.

Expand the Group Membership area (Optional):

6. In the Available Groups pane, select a group to which you want this Player Machine added, then click Add.
7. The selected groups are moved to the Member of Groups pane.
8. Close the Group Membership.

Note: You also can add a Player to a Group by dragging and dropping the Player from the list in the Players area to the appropriate Group in the Groups area, or by dragging and dropping a Group in there Groups area to a Player int he Players area.

In the ApplicationVantage Settings area:

Note: The fields on this tab are available only if ApplicationVantage is installed on the Player Machine and ApplicationVantage Mode is selected when you choose a script in the Script Assignment tab.

From the drop-down list in the NIC Name field, select the Network Interface Card (NIC) that is used by the machine, if necessary.

To save the Player machine:

Click Save, then click OK. The Player Machine appears in the Manage Players and Groups dialog box in the All Player Machines and Groups tree.

Saving Machine Configurations

After configuring the machines to use for a load test, you can save the machine configuration information into a configuration file (.cfg) that can be reused in later tests. This saves you significant time setting up later tests. A configuration file includes information about which machines on the network were used as Player machines. You can save multiple configurations under different names. By default, when first using QALoad, the Conductor uses a configuration file named `Default.cfg`. The Conductor saves any changes to your machine configurations to this file unless you save your configuration to a new file with a different name.

You can open or save .cfg files from the Manage Players/Groups dialog box. The .cfg field always displays the active configuration.

To create a new, empty .cfg file:

1. On the Conductor toolbar, click Tools>Manage Players. The Manage Players/Groups dialog box displays.
2. Click File>New>Player or click the New icon on the toolbar.
3. In the Player Information area of the window, type a name in the Machine [hostname or IP] field.
4. Click File>Save as... On the Save As dialog box, specify a name for the new file and click Save.
5. Add the necessary Player and agent machines using the fields and buttons on the Manage Players/Groups dialog box. The machines you configure are saved automatically to the file you just created.

To rename the current .cfg file:

1. On the Manage Players/Groups dialog box, click File>Save As...
2. On the Save As dialog box, specify a name for the new file and click Save.
3. Make any necessary changes to the configuration. Your changes are saved automatically to the file you just created.

To open a previously created .cfg file:

1. On the Manage Players/Groups dialog box, click File>Open. The Open Machine Configuration dialog box displays.
2. Choose the .cfg file to open.

Note: The .cfg file only stores information about Player machines. It does not store information specific to a test, such as script names or settings. Test specific information is saved in the session ID file. A session ID file for a specific test saves the name of the .cfg file associated with that test, and opens it automatically when the session ID file is opened. You can change the .cfg file at any time without being concerned about the session ID file.

Removing a Script or a Player from a Test

To remove a script from a test:

1. Right-click on the script icon for the appropriate script, then select Remove Script.
2. In the confirmation dialog box, click Remove.

To remove all scripts from a test:

1. Click Actions>Remove All Scripts.

2. In the confirmation dialog box, click Remove.

To remove a single Player from a test:

1. In the script test setup node, right-click on the Player to remove, then select Remove Selected.
2. In the confirmation dialog box, click Remove.

To remove all Players from a test:

1. Click Actions>Remove All Players/Groups.
2. In the confirmation dialog box, click Remove.

Setting Script Properties

Enabling and Disabling ApplicationVantage Mode

Use the Script Properties panel to enable ApplicationVantage mode. This option is only available if ApplicationVantage is installed on the Player machine.

Note: Setup the ApplicationVantage settings using the Tools menu, then selecting Manage Players to open the Manage Players/Groups dialog box.

To enable ApplicationVantage mode from the Script Properties panel:

Click the script icon for the appropriate script to display the Script Properties panel on the right-had side of the window. In the Script Properties panel, click the ApplicationVantage Mode field, then use the arrow key to enable or disable ApplicationVantage mode. Selecting:

! True enables ApplicationVantage mode

! False disables ApplicationVantage mode

Setting External Data Options

Use the external data options to add or remove attached files, specify a central datapool file, or specify local datapool files used by your script. You can setup external options using the Script Properties panel in the Visual Designer, or using the Grid View window.

To assign a Local Datapool file used by the script:

Note: A local datapool file must reside in the directory \QALoad\Datapools on the Player workstation.

1. Do one of the following:

   ! In the Visual Designer, click the script icon for the appropriate script to display the Script Properties panel on the right-hand side of the window, then click the browse [...] button in the Local Datapools field. The Script Properties dialog box appears with the Attached Files window displayed.

OR

- ! In the Grid View window, select the Script Assignment tab, then click the browse [...] button in the External Data column. The Script Properties dialog box appears. Click Local Datapools.

2. Click Add. The Choose a Local Datapool File dialog box appears.
3. Select a file, and click Open.
4. Repeat steps 2 and 3 to include additional datapool files.
5. Click OK.

To assign attached files used by the script:

1. Do one of the following:
   - ! Open the Script Properties panel for the appropriate script, then click the browse [...] button in the Attached Files field. The Script Properties dialog box appears with the Attached Files window displayed.

OR

   - ! In the Grid View window, select the Script Assignment tab, then click the browse [...] button in the External Data column. The Script Properties dialog box appears. Click Attached Files.

2. Click Add. The Choose a file to attach dialog box appears.
3. Select a file, and click Open.

To assign a Central Datapool file used by the script:

1. Do one of the following:
   - ! Open the Script Properties panel for the appropriate script, then click the browse [...] button in the Central Datapool field. The Script Properties dialog box appears with the Central Datapool window displayed.

OR

   - ! In the Grid View window, select the Script Assignment tab, then click the browse [...] button in the External Data column. The Script Properties dialog box appears. Click Central Datapool.

2. Click Browse. The Choose a Central Datapool File dialog box appears.
3. Select a file, and click Open.
4. (Optional) Select Rewind to rewind the records from this datapool at the end of a test. This enables you to reuse the records in a subsequent test of this session.
5. (Optional) Select Strip to remove the datapool records from the test so that they cannot be used again.

Setting Middleware Options for Citrix and SAP

You can set the following custom options for Citrix and SAP middlewares:

Middleware Type	Option	Description
Citrix	Hide Graphical User Interface for Citrix Users	Runs Citrix in "windowless" mode.

SAPGUI	Hide Graphical User Interface for SAP Users	Runs SAP in "windowless" mode.

You can setup Citrix and SAP middleware options using the Script Properties panel in the Visual Designer or in the Grid View window.

To set the middleware options in the Visual Designer's Script Properties panel:

1. Click the appropriate Citrix or SAP script icon 📜 . The Script Properties panel appears on the right-hand of the window.
2. Select the Hide Citrix/SAP graphical user interface field, and use the arrow key to select the desired middleware option. You can select:
   - ! True - to hide the graphical user interface
   - ! False - to display the graphical user interface
3. Click OK.

To set the middleware options in the Grid View window:

1. In the Script Assignment tab, select the appropriate Citrix or SAP script.
2. In the Middleware column, click the browse [...] button. The Middleware Options dialog box appears.
3. Select Hide graphical user interface during replay to run the script in windowless mode.
4. Click OK.

Setting Debugging Options for a Script

If you encountered errors while validating or testing a script, use QALoad's debugging options to monitor the Player(s) that generated errors while they are running or after the test.

You can watch a virtual user execute a script on a Player Workstation while it is running. To monitor selected virtual users at runtime, enable the Debug Trace option before you run your test. Each virtual user for which you enabled Debug Trace displays messages on its assigned Player workstation indicating which commands are being executed.

You can instruct the Conductor to generate and save details about the script execution of selected virtual users by enabling Logfile Generation before you run your test. This applies to Citrix, ODBC, Oracle, Oracle Forms Server, SAP, Winsock, or WWW only.

You can set the Debug Options using the Script Properties panel in the Visual Designer, or using the Grid View window.

To enable the Debug options:

1. Do one of the following:

- In the Visual Designer, click the script icon for the appropriate script to display the Script Properties panel on the right-hand side of the window, then click the browse [...] button in the Debug Options field. The Debug window of the Script Properties dialog box appears.

OR

- In the Grid View window, select the Script Assignment tab, then click the browse [...] button in the Debug Options column, for the appropriate script. The Debug window of the Script Properties dialog box appears.

2. On the Debug Options dialog box, you can choose the following options:
   - To enable the Debug Trace option: in the Debug Trace Virtual User Range area, choose which virtual users (if any) to monitor. You can choose None or All Virtual Users, or choose Virtual User(s) and then type the numbers assigned to the virtual users you want to monitor. You can monitor individual virtual users or ranges of virtual users.
   - To enable Logfile Generation: in the Logfile Generation Virtual User Range area, choose which virtual users (if any) to monitor. You can choose None or All Virtual Users, or choose Virtual User(s) and then type the numbers assigned to the virtual users you want to monitor. You can monitor individual virtual users or ranges of virtual users.
3. Click OK to save your changes.
4. From the Conductor's main menu, click File>Save to save your test session ID.
5. Run your test as usual.

Note: Some log files are generated automatically when you run a test in the Script Development Workbench or Player.

Setting Error Handling Options

You can configure the behavior of a virtual user when an error occurs during the load test so that the test aborts or continues executing. You can set the Error Handling Options using the Script Properties panel in the Visual Designer, or using the Grid View window.

To set the Error Handling options:

1. Do one of the following:
   - In the Visual Designer, click the script icon for the appropriate script to display the Script Properties panel on the right-hand side of the window, then click the browse [...] button in the Error Handling field. The Error Handling page of the Script Properties dialog box appears.

   OR

   - In the Grid View window, click the Script Assignment tab, then click the browse [...] button in the Error Handling field for the appropriate script . The Error Handling page of the Script Properties dialog box appears.
2. Select the option you want applied to the script. You can select:
   - Abort, stopping further execution of transactions. Use this option when errors will make the virtual user invalid for executing more transactions.
   - Continue executing and ignore the error. Select this option when errors are not critical to the performance of the load test

- (WWW, SAPGUI, and Citrix scripts only) Restart the transaction from the beginning. When you select this option, you must type a number for the attempts made to restart the transaction in the Maximum restart attempts field.

Note:The transaction count increases for each transaction that is restarted.

3. (Optional) Select Apply Error Handling to all scripts in this session. This ensures that all scripts in the session respond to errors the same way.

Setting Script Pacing

Script Pacing is the time interval between the start of a transaction and the beginning of the next transaction on each workstation running the script. For example: if a transaction is designed to duplicate the process of someone handling incoming telephone calls and those calls arrive at a rate of 40 per hour/per person, set the pacing rate at 90 seconds. The default pacing value is one second, which enables the Conductor to control runaway virtual users.

Note: QALoad randomly schedules transactions so that each transaction executes on an average according to this predetermined rate. When a transaction completes faster than its pacing rate, QALoad delays the execution of the next transaction for that workstation so that proper pacing is met. Since we do not normally time events according to this predetermined rate, QALoad randomly accelerates or delays the pacing on a workstation-by-workstation basis. However, on the average, QALoad provides pacing according to the value that you assign.

You can set the Pacing rate using the Script Properties panel in the Visual Designer, or using the Grid View window.

To set the Pacing rate:

1. Do one of the following:

   - In the Visual Designer, click the script icon for the appropriate script to display the Script Properties panel on the right-hand side of the window, then click the browse [...] button in the Pacing field. The Set Script Pacing dialog box appears.

   OR

   - In the Grid View window, click the Script Assignment tab, then click the browse [...] button in the Pacing field. The Set Script Pacing dialog box appears.

2. Use the arrows in each field to set the pacing rate. You can set Hours, Minutes, Seconds, and Milliseconds.
3. Click OK.

Setting the Service Level Threshold

Use the service level threshold to specify a response time that is used to compare against incoming response time data. The threshold appears as a horizontal line in the runtime Graphs view.

Service level thresholds are similar to thresholds that can be specified for real-time graphs in the Conductor, but are limited to transaction time and must be specified before a test runs.

Set the Service Level Threshold in the Visual Designer's Script Properties panel.

To set the service level threshold:

1. In the Visual Designer, click the script icon for the appropriate script to display open the Script Properties panel on the right-hand side of the window.
2. In the Service Level Threshold field, click the browse [...] button to open the Set Service Level Threshold dialog box.
3. Use the arrow keys in the Hours, Minutes, and Seconds fields to set the threshold.
4. Click OK.

Setting the Sleep Factor Percentage

Use the Sleep Factor % field to specify the percentage of any originally recorded delay to preserve in the script. QALoad records the actual delays between requests and inserts the DO_SLEEP command in the script to mimic those delays when the script is played back in a test. You can maintain the exact length of the recorded delays at playback, or shorten them by entering a smaller percentage of the originally recorded delay to play back. For example, if you recorded a delay of 10 seconds, then DO_SLEEP (10); is written to your script. Then, if a Sleep Factor of 50% is specified here, the Player sleeps for 5 seconds at that statement when the test is executed.

Valid values for Sleep Factor % are 0-1000%, and Random. Random causes the Player to sleep for a randomly selected duration between 0 and 100. A value of 100% causes the script to execute at exactly the same speed at which it was recorded; therefore, you can simulate the performance of faster users by specifying a lower Sleep Factor % value.

Hint: Enter a value of zero during unit testing to eliminate the actual sleeps from the script. After you unit test the script, you can restore the original recorded delays by changing the Sleep Factor to a higher percentage.

You can set the sleep factor percentage using the Script Properties panel in the Visual Designer, or using the Grid View window.

To set the Sleep Factor percentage:

1. Do one of the following:

   ! In the Visual Designer, click the script icon for the appropriate script to display the Script Properties panel on the right-hand side of the window.

   OR

   ! In the Grid View window, click the Script Assignment tab, then click the Sleep Factor % field for the appropriate script.

2. In the Sleep Factor % field, do one of the following:

   ! Click the arrow in the field to select Random or 100

   OR

   ! Type the number representing the percent of the originally recorded delay to keep in the script.

3. In the Conductor, click File>Save.

Setting Options for Large Amounts of Timing Data

Your load test probably includes a large number of checkpoints and virtual users in order to adequately test your system. When your test is running and your Conductor is collecting timing information from your Player machines, the sheer amount of data can take up more of your resources than you'd like to expend.

Use QALoad's Timing Data Thinning option to thin the amount of timing data being transferred back to the Conductor during the test so that your test can run longer without stressing your resources.

You can set the Timing Options using the Script Properties panel in the Visual Designer, or using the Grid View window.

To open the Timing Options dialog box:

! In the Visual Designer:

1. Click the script icon for the appropriate script to display the Script Properties panel on the right-hand side of the window.
2. In the Timing Options field in the Script Properties panel, click the browse [...] button. The Timing page of the Script Properties dialog box appears.

! In the Grid View window:

1. Click the Script Assignment tab.
2. In the Timing Options column, click the browse [...] button. The Timing page of the Script Properties dialog box appears.

To set timing options:

In the Timing Options area of the dialog box, select options for checkpoints and for custom counter data collection. For more information on these options, refer to Timing Options.

To thin timing data:

1. In the Timing Data Thinning area of the dialog box, choose one or both of the following:

   ! Thin counter timing data by to control the amount of counter timing data that is collected and saved in the timing file. Do not select this option if you want to collect all available timing data for counters.

   ! Thin checkpoint timing data by to control the amount of checkpoint timing data that is collected and saved in the timing file. Do not select this option if you want to collect all available timing data for checkpoints.

2. Do one of the following:

   ! Select Script to thin data by script. In the Summary interval field, type the number of seconds between each data collection.

   ! Select Virtual User to thin data by virtual user. In the Summary interval field, type the number of seconds between each data collection.

   Note: Thinning by script minimizes the amount of data collected.

3. The average is sent to the Conductor for inclusion in the timing file, rather than every value.
4. Click OK.
5. Save your changes to your test session ID file by choosing File>Save from the Conductor main menu.

Setting the Number of Transactions

Use the Transactions field to set the number of transactions that each Virtual User running the script should run. Once the workstation executes the number of transactions that you specify, script execution continues with the line following the End Transaction command rather than jumping to the beginning of the transaction loop.

You can set the number of transactions using the Script Properties panel in the Visual Designer, or using the Grid View window.

To set the number of transactions in the Visual Designer:

1. Click the script icon for the appropriate script. The Script Properties panel displays on the right-hand side of the window.
2. In the Transactions field, type the number of transactions that each Virtual User should run.
3. From the Conductor menus, click File>Save.

To set the number of transactions in the Grid View window:

1. Click the Script Assignment tab.
2. In the Transactions column for the appropriate script, use the arrow keys to select the number of transactions that each Virtual User should run.
3. From the Conductor menus, click File>Save.

Setting Player Properties

Specify Virtual User Configurations

Once you've selected the scripts and configured the transaction settings, you must assign Player machines and groups, and set the Virtual User configurations. You can set the Virtual User configurations using:

- the toolbar in the script test setup node in the Visual Designer
- the Player Properties panel in the Visual Designer
- the Machine Assignment tab in the Grid View
- the toolbar in the main Visual Designer window

To set the Virtual User configurations in the script test setup node:

1. Click the appropriate Player in the script test setup node in the Visual Designer.
2. Click the Configure All button in the script test setup node toolbar. The Configure All Players/Groups dialog box appears.
3. Click the arrows in the fields to set the following options:
   - Starting VUs - Number of Virtual Users to launch the script on the machine when the test begins.
   - Ending VUs - Number of Virtual Users at the end of the test.

- ! VU increment - Number of virtual users to add at intervals throughout the test.

Note: If you add incremental virtual users, you must designate the time interval and the ending virtual users.

- ! Time interval - Time interval at which incremental virtual users should be added to the test.
- ! Mode - The test mode for the machine. You can select thread based or process based.

4. To use these options for all players and groups in the script, click Assign to all selected players and groups.
5. Click Finish.
6. Click File>Save in the Conductor main menu.

To set the Virtual User configurations using the Player Properties panel of the Visual Designer:

1. Click the individual Player in the script test setup node in the Visual Designer. The Player Properties panel displays on the right-hand side of the window.
2. In the [Starting VUs] field, type the number of Virtual Users to launch the script on this machine when the test begins.
3. (Optional) In the [VU Increment] field, type the number of virtual users that should be added at intervals. When you fill in this field, you must also fill in the Time Interval and Ending VUs fields.
4. In the Ending VUs field, type the number of Virtual Users at the end of the test.
5. In the Mode field, use the arrow to select the test mode, process-based or thread-based, for this Player.
6. (Optional) In the Time Interval field, click the browse [...] button to display the Set Time Interval dialog box.

Note: If you filled in the VU Increment field, you must fill in the Time Interval.

7. Use the arrows in the Hours, Minutes, and Seconds fields to set the time interval at which incremental Virtual Users should be added to the test, then click OK.
8. Click File>Save in the Conductor main menu.

To set the Virtual User configurations using the Grid View:

1. In the Machine Assignment tab, click the down arrow in the Player Name field, then select a player from the list.
2. (Optional - WWW only) In the Middleware field, click the browse [...] button to open the Expert User Options dialog box. If you enable the Expert User, select the Virtual User number to represent the Expert User.
3. Click each of the following fields to set options for:
   - ! Starting VUs - Type the number of Virtual Users to launch the script on the machine when the test begins.
   - ! Ending VUs - Type the number of Virtual Users at the end of the test.
   - ! VU Increment - Use the arrows in the field to set the number of virtual users to add at intervals throughout the test.
   - ! Timing Interval - Click the browse [...] button in the field to open the Set Time Interval dialog box. Set the time interval at which incremental virtual users should be added to the test.

Note: If you add incremental virtual users, you must designate the time interval and the ending virtual users.

- Mode - Use the arrows in the field to select the test mode for the machine. You can select thread based or process based.

4. Click File>Save in the Conductor main menu.

To set the Virtual User Configurations using the Visual Designer toolbar:

1. Select the appropriate script test setup node, then click Review Virtual Users in the Visual Designer toolbar.
2. Use the columns in the Review Virtual Configuration dialog box to set the Virtual User configurations for each Player listed.

Enabling Expert User

When you enable the Expert User, this VU collects more detailed information about requests that are made while the script is running. Every main request and subrequest logs the amount of server and network time used. This helps diagnose why page loads may be taking longer than expected. You enable Expert User from the Conductor, either before or during a load test. Expert User uses the existing custom counter support so Conductor can graph the custom counter information. Once the load test is complete, you can view the data in Analyze.

Note: Currently, Expert User capability is provided only for the WWW middleware.

You can enable the Expert User:

Before a load test begins from the Visual Designer or Grid View windows.

During a load test from the Runtime window.

To enable Expert User before the load test begins:

1. Do one of the following:
   - Using the Visual Designer:
     a. Click an individual player machine in the script test setup node to display the Player Properties panel on the right-hand side of the window.
     b. Click the browse [...] button in the Expert User Options field to open the Expert User Options dialog box.

OR

   - Using the Grid View window:
     a. Click the Machine Assignment tab.
     b. In the Middleware field for the appropriate WWW script, click the browse [...] button to display the Expert User Options dialog box.
2. Click Enable Expert User timings.
3. In the Virtual User Number field, type the Virtual User (VU) number to represent the Expert User. The default VU number is zero (0).
4. Click OK.

To enable Expert User during the load test:

1. In the Runtime window, click Actions>Set Expert User Options. The Update Expert User Options dialog box displays listing all the scripts that support Expert User counters.
2. Click the scripts in which you want to enable Expert User, then click OK.

Note: Selecting Expert User Scripts at the top level enables or disables expert user on all associated scripts.

Changing the Number of Virtual Users

Change the number of virtual users assigned to a script using the Visual Designer or on the Machine Assignment tab of the Grid View window.

To change the number of virtual users using the Visual Designer:

1. Do one of the following:

   ! In the Visual Designer window, select a player machine in the appropriate script node. The Player Properties panel displays on the right-hand of the window. Type a new value in the Starting VUs column of the Player Properties panel.

   OR

   ! Click Review Virtual Users in the Visual Designer toolbar, then type a new value in the Starting VUs column for the selected script. Click OK.
2. If you have assigned incremental virtual users, change the values in the VU Increment column and the Ending VUs column to determine how many virtual users to add at the interval specified in the Time Interval column.
3. Select File>Save to save your changes to the current session ID file, or File>Save As to save them to a new session ID file.

To change the number of virtual users using the Grid View:

1. In the Machine Assignment tab, type a new value in the Starting VUs column for the selected script.
2. If you have assigned incremental virtual users, change the values in the VU Increment column and the Ending VUs column to determine how many virtual users to add at the interval specified in the Time Interval column.
3. Select File>Save to save your changes to the current session ID file, or File>Save As to save them to a new session ID file.

Using the Test Configuration Wizard

About the Test Configuration Wizard

The Test Configuration Wizard consists of a series of screens that guide you through the process of setting up a load test. Use the Test Configuration Wizard to select scripts, configure the script's transaction settings, select players or groups to assign to the test, and specify desired Virtual User configurations. You also can add scripts to an existing session using the Test Configuration Wizard. Once you setup your load test with the Test Configuration Wizard, you can run the test immediately without additional configuration.

You can use the Test Configuration Wizard to:

Create a New Test Session Using the Test Configuration Wizard

Modifying an Existing Test Session Using the Test Configuration Wizard

Creating a Test Session Using the Test Configuration Wizard

The Test Configuration Wizard guides you through the process of setting up a load test. Use the series of screens in the Test Configuration Wizard to:

1. Select scripts
2. Configure the script's transaction settings
3. Select players or groups to assign to the test
4. Specify desired Virtual User configurations

You can open the Test Configuration Wizard from the Conductor Start page or with a test session open in the Conductor.

To access the Test Configuration Wizard from the Conductor Start page:

1. Do one of the following:
   - In the Tasks area, click Test Configuration Wizard to open the first screen of the wizard.

   OR

   - In the Recent Sessions area, click New to open the Create New Session dialog box. Select the Launch Test Configuration Wizard option.

   Note: If you do not select the Launch Test Configuration Wizard option, the new test session opens in Conductor's Visual Designer, where you can manually configure the load test session.
2. In the Name field, type a name for the new session.
3. (Optional) In the Description field, type a description for the new session.
4. Click Next to start configuration of your test session.

To access the Test Configuration Wizard with a session open in Conductor:

Click File>New, then follow steps 2 through 4 above.

Modifying a Test Session Using the Test Configuration Wizard

You can use the Test Configuration Wizard to add a script to an open test session by following the steps for each screen in the Test Configuration Wizard.

To start the process of adding a script using the Test Configuration Wizard:

On the toolbar, click the Test Configuration Wizard button. The Select Script to Configure dialog box appears.

Anticipating Error Conditions

You know before beginning a load test that errors are a possibility, but you may not always want them to stop your progress during testing.

QALoad helps anticipate error conditions and determine, before running the test, how Players react to non-fatal errors. By setting one option, you can instruct a Player to continue as if no error was encountered, stop running immediately, or restart at the beginning of the transaction.

Note: When the Conductor process stops for any reason during a load test, the associated Players automatically terminate.

Managing Large Amounts of Test Data

With a large number of virtual users, it is possible to create a timing file containing hundreds of thousands of timing records for each checkpoint. Attempting to graph just a few of those checkpoints can slow down QALoad Analyze considerably.

For example, if a timing file contained 250,000 timing records for each data point, attempting to graph even one checkpoint means that QALoad Analyze must paint 250,000 lines on the graph. Since most monitors only have 1024 pixels across the screen, the 250,000 data points would mostly be plotted atop one another and the results would be unreadable.

Now imagine attempting to graph the data of several data points of that size. The sheer amount of data could easily overwhelm a workstation. And every time you move the window, resize the window, or right-click on the graph, QALoad Analyze has to re-draw the graph. You could conceivably spend enormous amounts of time simply attempting to graph data.

To make large amounts of data manageable, QALoad Analyze provides an option that allows you to determine how to thin data. That is, how to determine how many data points to plot.

When your test is running and your Conductor is collecting timing information from your Player machines, the sheer amount of data can take up more of your resources than you would like to expend. Use QALoad's Timing Data Thinning option to thin the amount of timing data being transferred back to the Conductor during the test so that your test can run longer without stressing your resources.

Validating Scripts in Conductor

Before running a test, you should run your script in a simple test to ensure that it runs without errors. You can validate UNIX or Win32 scripts in the Conductor.

Managing Players and Groups

Overview of Players and Groups

You can configure the various machines and agents that will participate in a load test from a single screen. Click Tools>Manage Players to display the Manage Players/Groups dialog box, where you can configure Player Machines, Player Groups, and ApplicationVantage settings information from a single screen.

You should use this option to update Player or Agent information whenever a Player or Agent is added to the test network, removed from the test network, or the network address of a Player or Agent has changed.

You can collect Player machines into logical groups using the Group Membership options in the dialog box.

Player Agents

Player machines execute the virtual users that perform the transactions recorded in your test scripts. You can view information on Player machines from either the Visual Designer or the Grid View window. If no Player machines are listed, you can retrieve information from Player machines on the local network, or you can add Player machines manually.

Managing Player Machines

Adding Player Machines to a Test Session

Follow these instructions to add a Player workstation to your pool of available Players in a test's session ID file.

To add a new Player machine:

1. From the Conductor's main menu, click Tools>Manage Players. The Manage Players/Groups dialog box displays.
2. Click File>New>Player. A new page displays in the detail area of the dialog box, where you enter information and settings for the new Player.

To enter information and settings for the new Player:

Enter information for the new Player in the following areas of the dialog box.

In the Player Information area:

1. In the Machine (hostname or IP) field, type a name for the Player Machine.
2. In the Communications [TCP] port field, type the port number the Conductor should use to communicate (using TCP) with this machine during a test. The default is 3032.
3. Click Verify to check that the machine is active. The Player or Agent returns the operating system, processor type, amount of memory, and the maximum threads and processes available on the machine.

Note: If a Player does not respond, a message box appears indicating that the Player is not responding. If the Player is not responding, one of the following scenarios is likely:

! The host name and/or port number you entered may not be correct. Check your parameters and network connections, then try to send another request.

! The Player is not running. Start the Player and then try to send another request.

Expand the Machine Settings area (Optional):

4. Select desired options to ping the host before attempting connection to the player, generate IP spoof data, or override the default machine settings.
5. Close the Machine Settings.
6. Expand the Group Membership area (Optional):
7. In the Available Groups pane, select a group to which you want this Player Machine added, then click Add.
8. The selected groups are moved to the Member of Groups pane.
9. Close the Group Membership.

Note: You also can add a Player to a Group by dragging and dropping the Player from the list in the Players area to the appropriate Group in the Groups area, or by dragging and dropping a Group in there Groups area to a Player int he Players area.

In the ApplicationVantage Settings area:

Note: The fields on this tab are available only if ApplicationVantage is installed on the Player Machine and ApplicationVantage Mode is selected when you choose a script in the Script Assignment tab.

From the drop-down list in the NIC Name field, select the Network Interface Card (NIC) that is used by the machine, if necessary.

To save the Player machine:

Click Save, then click OK. The Player Machine appears in the Manage Players and Groups dialog box in the All Player Machines and Groups tree.

Editing a Player Machine

From the Conductor's main menu, click Tools>Manage Players. The Manage Player Machines and Groups dialog box displays. Use the following procedure to edit Player Machines.

To edit a Player machine:

Select an individual Player Machine in the Players list.

In the Player Information area:

1. In the Communications port field, type the port number the Conductor should use to communicate (using TCP) with this machine during a test. The default is 3032.
2. Click Verify to check that the machine is active. The Player or Agent returns the operating system, processor type, and amount of memory on the machine.

In the Machine Settings area:

1. Open the Machine Settings area.
2. Make any necessary selection or changes here, then close the Machine Settings.

In the Group Membership area (Optional):

1. Open the Group Membership area.
2. Add the Player to appropriate groups by selecting a group in the Available Groups pane and clicking Add.
3. Close the Group Membership.

In the ApplicationVantage Settings area:

Note: These fields are enabled only if Application Vantage is installed on the Player machine.

! In the NIC Name field, select the Network Interface Card (NIC) that is used by the machine.

Save the edits to the Player machine:

! Click Save to save your settings for this player machine, then click OK.

To delete a Player machine:

Select the Player machine in the Players panel, then click Edit>Delete.

Discovering and Verifying Player Machines

In Conductor, you can retrieve information from Player machines on the local network by doing the following:

To discover and information on Player machines:

1. Click Tools>Manage Players. The Manage Players/Groups dialog box displays.
2. Click Actions>Discover Players. QALoad Conductor queries the network for available Player workstations and adds the results under Player Machines in the Players area.

To verify a Player machine:

1. In the Players area of the screen, select the Player machine to verify.
2. In the Communications [TCP] port field in the Player Information area, click Verify. A confirmation dialog box appears when the verification is successful.

Viewing Information on Player Machines

In Conductor, you can retrieve information from Player machines on the local network by doing the one of following:

To view information on Player machines in the Visual Designer:

In the script node, select the individual Player you want to view. The information on the Player appears in the Properties panel on the right-hand side of the window.

To view information on the Player in the Grid View:

Click View>Grid Window to display the Grid View window, then click the Machine Assignment tab to display information on the Player.

Managing Groups

Adding a Group

You can combine players into logical groups using the following procedure:

To create a group:

1. From Conductor's main menu, click Tools>Manage Players. The Manage Players/Groups dialog box displays.
2. Click File>New> Group. The Group Information dialog box appears.
3. In the Name field, type a name for the group.
4. In the Description field, type a description for the group.

To add Players to the group:

1. In the Available Players panel, select the player or players to add to the group.
2. Click Add. The Player is moved to the Member Players pane.

Note: You can select more than one machine by holding down the Ctrl key and selecting each Player Machine to select. Select all the available Player Machines by clicking Add All.

3. Click Save, then click OK.

Editing a Group

Use the following procedure to edit a group.

To edit a group:

1. On Conductor's main menu, select Tools>Manage Players. The Manage Players/Groups dialog box displays.
2. Select a group in the Groups panel to display the Group Information window.
3. Use the fields in this dialog box to change the Group Name or Description.
4. To add a Player Machine to the group, use the procedures for adding a Player machine to a group.

To remove a Player from the group:

1. Select the Player machine in the Member Players panel, then click Remove.
2. Click Save to save your changes, then click OK to return to the main Conductor window.

Running a Test

Running a Load Test

After validating a script, it is safe to run a load test with that script.

To start a load test:

In the Conductor, click the Run button on the configuration and setup toolbar, or from the Actions menu, choose Run. While a test is running, the Conductor's Interface changes to provide you with real-time test options. For more information, see Runtime Window Interface.

Note: While any window on the desktop is re-sizing or re-positioning, all Windows applications pause. Do not click and hold on a window caption or border for extended periods during a load test because it delays message handling and may have an impact on test results.

While a load test is running, the Conductor's toolbar changes from the Configuration and Setup Toolbar to the Runtime Toolbar. The Runtime Toolbar buttons let you control the test and access detailed information about the test while it is running. For more information, see Monitoring a Load Test. This gives detailed information about what to expect from the QALoad Conductor while a test is running — including descriptions of the Runtime Toolbar buttons

Running a Series of Tests

You can also run a series of tests — a batch test. A batch test comprises multiple session ID files. When you run a batch test, the session files are executed sequentially until all of them are executed. The Conductor enables you to run multiple batch tests without operator intervention. For more information, see Running a Batch Test.

Checking Out Virtual User Licenses

If you are licensed to run multiple copies of the Conductor, for example so different work groups have access to QALoad, you can check out virtual user licenses before running a load test to ensure that enough are available for your test run.

If you do not choose to check out your licenses before starting a test, QALoad prompts you after you start the test and attempts to check out the appropriate number of licenses. We recommend that you check your licenses out manually before starting so you can be sure you have enough virtual users available before beginning your test run.

To check out virtual user licenses:

1. From the Conductor menu, select Tools>Licensing. The License Information dialog box appears.
   - If you are licensed for concurrent licensing (multiple Conductors) the Conductor queries your license server to determine how many licenses are currently available, and returns the results to this dialog box. Go to step 2.
   - If you have a node-locked license (a single Conductor), then most of the options on this dialog box are unavailable, as you will not need to, or be able to, check out virtual user licenses. All virtual users for which you are licensed are available only to this Conductor. Click OK to return to your test setup.
2. In the Licensing Operations area, select Check Out Virtual User Licenses, then type how many virtual user licenses you want to check out in the Number of Licenses field.
3. Click Check Out. The licenses are checked out to your Conductor, and are unavailable to any other Conductor workstations on the network.

When you are done using your licensed virtual users, check them back in so they are once again available to other Conductor workstations on your network.

To check in virtual user licenses:

1. From the Conductor menu, choose Tools>Licensing. The License Information dialog box appears.
2. If you have licenses checked out, the Check in Virtual User License option is automatically selected for you.
3. Click Check In. The licenses are made available to other Conductor workstations on the network.

Dialing Up/Down Virtual Users

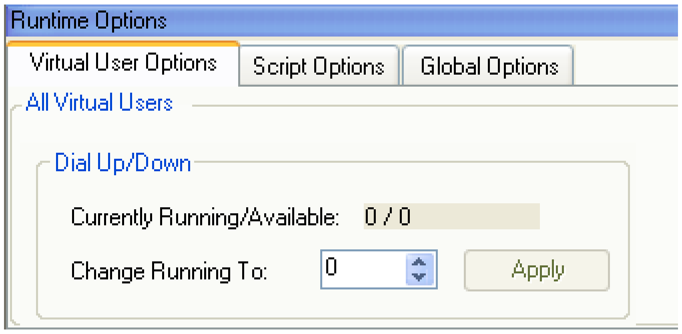
QALoad's dial-up/dial-down feature in Conductor allows you to dynamically add or reduce virtual users to your test at the script or Player level while your test is running. This enables you to adjust your running test according to test behavior on-the-fly, rather than stopping to re-configure playback criteria.

To use the dial-up/dial-down feature, you must:

- be licensed for at least the number of virtual users requested

- configure a dial-up/dial-down session before running the test
- If you have not configured a dial-up/dial-down session, you will not be allowed to add or suspend virtual users while the test is running. For more information, see Configuring a Dial-up Session.
- Dial-up/dial-down is enabled only after all Virtual Users configured for the test session are ramped up.
- Dial-up/dial-down is not supported for a machine assignment entry that is using a player group.

When your test is running, the bottom of the Test Information window turns into the dockable Runtime Options panel, a portion of which is shown below:



Currently Running/Available: 0 / 0

Change Running To: 0

When you click on a Player or script in the test's tree-view, the Runtime Options panel indicates how many virtual users are currently running on the selected Player machine or script. You can change the number of running virtual users per script or per Player by selecting the appropriate script or Player machine in the tree-view, and then typing a new number in the Change Running To field (or by using the up/down arrows).

To dial up or down (add or subtract) virtual users during a test:

1. When your test is running, click on the script or Player workstation in the Runtime Window's tree-view for which you want to add or subtract virtual users. On the Virtual User Options tab, the Currently Running/Available field shows how many virtual users are currently running on that script or Player.
2. In the Change Running To field, type a new number or use the up/down arrows to change the number.
3. When you are done, click Apply. The Conductor will release or suspend the specified number of virtual users.

Notes:

- Your changes do not take effect until you click Apply.
- When you dial down a virtual user, the virtual user finishes the current transaction before going into a suspended state.

Increase/Decrease Runtime Timing Updates

While a test is running, you can change the frequency at which timing updates are sent from the Players to the Conductor in the Runtime window. Decreasing the update interval reduces the amount of overhead incurred in large load tests due to the communications between the Conductor and large numbers of virtual users.

To change the Runtime Timing updates:

1. On the Global Options tab of the Runtime Options panel (bottom pane), choose from the following options:
   - No Updates: Choose this option to stop sending timing data while the test is running. Data will still be collected at the end of the test.
   - Send All: Choose this option to send all timing data as it is compiled.
   - Periodic Updates: Choose this option to specify a time interval for sending updates, then type the time interval (in seconds) below.
2. Click Apply. Any change takes effect immediately, and applies to all scripts in the test.

Removing a Script or a Player from a Test

To remove a script from a test:

1. Right-click on the script icon for the appropriate script, then select Remove Script.
2. In the confirmation dialog box, click Remove.

To remove all scripts from a test:

1. Click Actions>Remove All Scripts.
2. In the confirmation dialog box, click Remove.

To remove a single Player from a test:

1. In the script test setup node, right-click on the Player to remove, then select Remove Selected.
2. In the confirmation dialog box, click Remove.

To remove all Players from a test:

1. Click Actions>Remove All Players/Groups.
2. In the confirmation dialog box, click Remove.

Removing Used Datapool Records After a Test

You can remove used datapool records from a Central datapool after a test completes by setting the Strip Datapool function before you run the test. Use this function when running a test where you have data in the external datapool that can only be used once by one virtual user at a time. (For example, when running transactions that have unique data constraints.) When activated, the Strip Datapool function marks each piece of data in the datapool that is used during your test. When the test is over, the Strip Datapool function prompts you to remove the identified used data from the datapool. If you run the test again, only new data is used for your subsequent test.

To use the Strip Datapool function:

1. With the current test's session ID file open, do one of the following:

- Click the script icon for the appropriate script to display the Script Properties panel on the right-hand of the window. In the External Data area, click the browse [...] button in the Central Datapools field.

OR

- In Grid View window's Script Assignment tab, click the browse [...] button in the External Data field of the appropriate script. When the Script Properties dialog box appears, click Central Datapool.

2. In the Central Datapool dialog box, click the browse [...] button to select the Central Datapool file, then select the Strip check box. Click OK.
3. At the end of your test, a Strip Datapools prompt appears asking if you wish to go to the Strip Datapools screen. Click Yes.
4. The Strip Data Pool dialog box appears with the Strip option selected. Click the Strip button.
5. When you are finished, click Close.

Stopping a Load Test

A load test is complete when all virtual users exit. A virtual user automatically exits when one of the following occurs:

- A script encounters an EXIT command.
- A script completes its transaction loop.
- A QALoad function fails and Abort on Error is set in Error Handling

To stop a load test:

Click the Exit All Virtual Users button or click the Quit Current Test button. The Virtual User icon changes to and the message, "Session aborted by User", displays.

Adding Post-test Comments

If you selected the Display post test comments dialog option when you configured the Conductor, the Post Test Comments window opens when you click the Quit Current Test button. Type any comments, which are saved to the test’s Summary Report that you can view in QALoad Analyze.

Adding Post-test Comments

By setting the appropriate options when you configure the Conductor, you can add comments to a completed test. The comments appear in the test's Summary Report in QALoad Analyze.

To configure the Conductor for adding post-test comments:

1. Select Tools>Options. The Options dialog box appears.
2. In the Dialog section of the Conductor Sessions page , select Display post test comments dialog.
3. Click OK. The Conductor is now configured so that you can add comments when a test completes.

To add post-test comments:

1. In the Test Completed dialog box, click Exit Test.

   OR

   In the Runtime toolbar, click the Quit current test button. The Post Test Comments dialog box displays.

2. Type any comments in the dialog box, then click OK. Your comments are saved in the Post Test Comments field of the Summary Report in

3. QALoad Analyze.

Monitoring a Running Test

Monitoring CPU Usage

To help you monitor the impact of running a load test on a server, QALoad can collect data from selected Players about CPU usage during a load test. The statistics collected during the test are merged into the test's timing file so you can view them in QALoad Analyze after the test.

Note: During a load test, if the CPU idle time of your machine falls below 25%, check the individual processes on your machine. If the Players and virtual users are utilizing most of the active CPU time, you should use additional Player machines and fewer virtual users per Player to conduct your load test.

Watching a Script Execute

Use the Debug tab in the Conductor Runtime window to view the executing script. Note that it is possible that you will not see the execution of every statement. In order to minimize network traffic between the Conductor and the Players, the Player sends its script debug status to the Conductor once per second, so that the Player can execute several statements without sending a debug message to the Conductor.

To open the Debug tab:

Select a Player in the Runtime window, then click the Debug Virtual User toolbar button.

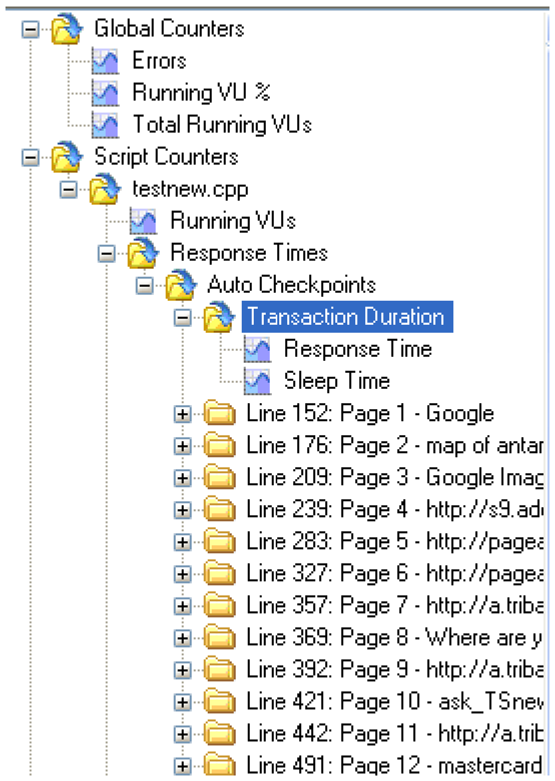
Note: The Conductor highlights the script line that it is currently executing.

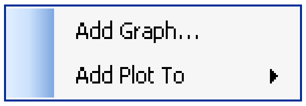
Graphing Checkpoints in Conductor

Use the Graphs view of the runtime Conductor to create real-time graphs of checkpoint response times during script execution.

Note: Similar graphs are also available for post-test analysis in QALoad Analyze.

Checkpoints are listed in the tree view on the left side of the Graphs view of the runtime Conductor, as shown in the example below. Both automatic and user-defined checkpoints appear in the Response Times folder of each running script.


Global Counters
Errors
Running VU %
Total Running VUs
Script Counters
testnew.cpp
Running VUs
Response Times
Auto Checkpoints
Transaction Duration
Response Time
Sleep Time
Line 152: Page 1 - Google
Line 176: Page 2 - map of antar
Line 209: Page 3 - Google Imag
Line 239: Page 4 - http://s9.ad
Line 283: Page 5 - http://pagea
Line 327: Page 6 - http://pagea
Line 357: Page 7 - http://a.triba
Line 369: Page 8 - Where are y
Line 392: Page 9 - http://a.triba
Line 421: Page 10 - ask_TSnew
Line 442: Page 11 - http://a.trib
Line 491: Page 12 - mastercard



Creating a Graph of Checkpoint Response Times

Before you can review checkpoint response times in graph form, you must select the checkpoint counters to include. To choose a checkpoint that should appear in a graph, highlight the checkpoint name, right-click and choose either Add Graph to create a new graph or Add Plot To to add a data plot to an existing graph.

If you choose the Add Graph option, the Add Graph dialog box appears. Select the options for how the graph should appear and click OK.

Adding Thresholds

To better identify problem checkpoints, you can set thresholds on plots or graphs that indicate the number of times the data record for that checkpoint has gone above or below the number you set. Thresholds can be set from the Advanced tab of the Add Graph dialog box or by right-clicking on an existing graph and choosing Thresholds.

Highlighting Individual Plots

If you create several plots on a single graph, it may become difficult to see individual plots. To increase a plot's visibility, click on a plot in the graph or a plot's number in the graph's legend. When highlighted, the plot appears thicker and darker on the graph.

Saving Checkpoint Graphs to a Session ID

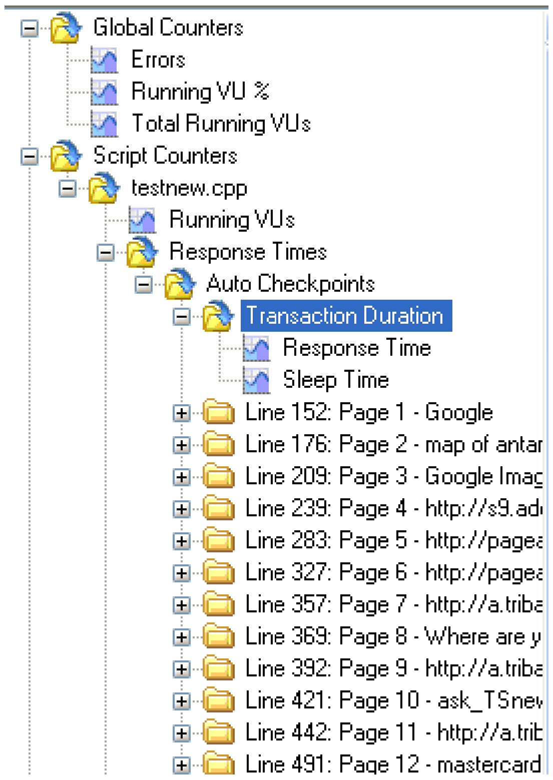
Checkpoint graphs that are created in the Conductor are automatically saved to the current session ID file. To remove all graphs you added, click Graph>Restore Default Graph Layout.

Graphing Counter Data

Use the Graphs view of the runtime Conductor to create real-time graphs of counter data during script execution. Similar graphs are also available for post-test analysis in QALoad Analyze.

Selecting Counters to Graph

All counter data that is available for graphing is located in the tree view on the left side of the of the Graphs view Data window, as shown below.



Global Counters
Errors
Running VU %
Total Running VUs
Script Counters
testnew.cpp
Running VUs
Response Times
Auto Checkpoints
Transaction Duration
Response Time
Sleep Time
Line 152: Page 1 - Google
Line 176: Page 2 - map of antar
Line 209: Page 3 - Google Imag
Line 239: Page 4 - http://s9.ad
Line 283: Page 5 - http://pagea
Line 327: Page 6 - http://pagea
Line 357: Page 7 - http://a.triba
Line 369: Page 8 - Where are y
Line 392: Page 9 - http://a.triba
Line 421: Page 10 - ask_TSnev
Line 442: Page 11 - http://a.trib
Line 491: Page 12 - mastercard

Scripts of any middleware type collect the following default counter data, which is available in the Conductor for real-time graphing:

- Global counters: Running VU%, total running VUs, and errors
- Script counters: Running VUs, response times, and transactions
- Player machine health: % processor, % memory used, % disk space, %disk time, % paging file

Additional middleware-based graphs are also generated by default and vary by middleware. For example, for the WWW middleware, several performance-based counters are automatically collected and available for graphing, including server responses and WWW traffic. You can monitor this data to determine the optimum rate of performance of the application that is running.

Graphing Counter Statistics

To choose a counter that should appear in a graph, highlight the checkpoint counter name or group of counters (folder), right-click and choose either Add Graph to create a new graph or Add Plot To to add a data plot to an existing graph.

If you choose the Add Graph option, the Add Graph dialog box appears. Select the options for how the graph should appear and click OK.

To better identify problems in the test, you can set thresholds on plots or graphs that indicate the number of times the data record for that counter has gone above or below the number you set. Thresholds can be set from the Advanced tab of the Add Graph dialog box or by right-clicking on an existing graph and choosing Thresholds.

Highlighting Individual Plots

If you create several plots on a single graph, it may become difficult to see individual plots. To increase a plot's visibility, click on a plot in the graph or a plot's number in the graph's legend. When highlighted, the plot appears thicker and darker on the graph.

Saving Counter Data Graphs to a Session ID

Counter data graphs that are created in the Conductor are automatically saved to the current session ID file. To remove all graphs you added, click Graph>Restore Default Graph Layout.

Running a Series of Tests (Batch)

Running a Batch Test

By setting the appropriate options in the Conductor, you can elect to run a series of tests as a batch, rather than one at a time. A batch test comprises multiple session ID files that are executed sequentially.

You can create a batch test by adding a number of session ID files to a batch file. Before you can add a session ID to a batch file, the following conditions must be true:

- The session must include a defined number of transactions. Sessions of unlimited transactions cannot be used in a batch test.
- All scripts to be included must exist before starting the batch test.

To run a batch test:

1. Select Actions>Batch Test. The Configure Batch Test dialog box appears.
2. Select the required session ID files in the Available Session Files list and click Add to add them to the Selected Sessions list.
3. If you want to run a previously defined batch, click the Load Batch File button in the toolbar to navigate to the directory where the batch file (.run) resides. Select it, and click OK.
4. In the Delay Between Tests field, click the up or down arrow to set the number of seconds to wait before starting the next test.
5. Click Save to save the current batch file, or click Save As... to save the batch file under a new name.
6. Click OK to return to the main Visual Designer window

OR

Click Start to begin running the batch test.

The Conductor then executes each of the session ID files in sequence.

Adding Sessions to a Batch Test

Before a session is added, the following conditions must be true:

! The session must include a defined number of transactions. Sessions of unlimited transactions cannot be used in a batch test.

! All scripts must exist prior to starting the batch test. This means that the files referenced in the selected session ID files are present in the script directory.

A session can be placed in a batch multiple times. This feature might be used to re-run a test or to perform housekeeping chores, such as logging users in or out of a host or database.

To add a session:

1. From the Actions menu, choose Batch Test. The Configure Batch Test dialog box displays.
2. In the Available Session Files box, highlight the session you want to add, and click the Add Add button.

If you want to run a previously defined batch, click the Load button to navigate to the directory where the batch file (.run) resides. Select it, and click OK.

The session is added to the Selected Sessions list on the right side of the dialog box.

3. In the Delay Between Tests field, click the up or down arrow to set the number of seconds to wait before starting the next test.
4. Click Save to save the current batch file, or click Save As... to save the batch file under a new name.
5. Click Start to begin running the batch test, or click OK to return to the main Visual Designer window.

Setting Delays Between Tests

You can set a fixed delay or pause between tests by specifying a value in the Delay Between Tests field on the Configure Batch Test dialog box. After each test is complete, the Conductor delays for the specified amount of time before starting the next test. To set up a series of tests, see Running a Batch Test.

Removing a Session from a Batch Test

To remove a session from a batch test:

1. Select Actions>Batch Test. The Configure Batch Test dialog box appears.
2. Click Load Batch File and select the file you want to modify.
3. In the Selected Sessions pane, highlight the session to remove and click Remove.
4. Click OK.

Terminating a Batch Test

Stop a batch of tests the same way you would stop a single session test, by clicking the Abort All Virtual Users or Exit All Virtual Users on the toolbar. The Virtual User icon changes to ✖ and the message, "Session aborted by User", displays. When the Conductor process stops for any reason during a load test, the associated Player processes automatically terminate.

Expert User

Overview of Expert User

Expert User provides an easy, logical guide for drilling down to the root performance problems for applications. It enables you to break web pages down into their individual components, providing more detailed response time data. Response time for each component is broken into network and server time.

More detailed information helps troubleshoot application performance problems. The ability to see timing files on a component level can spotlight where the majority of time is being spent. A breakdown of network and server times per component can identify areas for improvement in either the network or server hardware or configuration, or in application performance.

The main functionality is provided by a special virtual user (VU). When you enable the Expert User, this VU collects more detailed information about requests that are made while the script is running. Every main request and subrequest logs the amount of server and network time used. This helps diagnose why page loads may be taking longer than expected. For example, a particular subrequest, such as css, gif, html, and so forth, may be taking more time to download from the server than other requests. Expert User data can show you this. It also can help you determine whether the problem is a network or a server problem.

You enable Expert User from the Conductor, either before or during a load test. Expert User uses the existing custom counter support so Conductor can graph the custom counter information.

Once the load test is complete, you can view the data in Analyze. The Analyze Workspace includes an Expert User tab, from which you can access detail reports and graphs on server and network data. The pre-defined reports include an Expert User report.

Note: Currently, Expert User capability is provided only for the WWW middleware.

Enabling Expert User

When you enable the Expert User, this VU collects more detailed information about requests that are made while the script is running. Every main request and subrequest logs the amount of server and network time used. This helps diagnose why page loads may be taking longer than expected. You enable Expert User from the Conductor, either before or during a load test. Expert User uses the existing custom counter support so Conductor can graph the custom counter information. Once the load test is complete, you can view the data in Analyze.

Note: Currently, Expert User capability is provided only for the WWW middleware.

You can enable the Expert User:

Before a load test begins from the Visual Designer or Grid View windows.

During a load test from the Runtime window.

To enable Expert User before the load test begins:

1. Do one of the following:

   ! Using the Visual Designer:

   a. Click an individual player machine in the script test setup node to display the Player Properties panel on the right-hand side of the window.

   b. Click the browse [...] button in the Expert User Options field to open the Expert User Options dialog box.

   OR

- ! Using the Grid View window:
  a. Click the Machine Assignment tab.
  b. In the Middleware field for the appropriate WWW script, click the browse [...] button to display the Expert User Options dialog box.
2. Click Enable Expert User timings.
3. In the Virtual User Number field, type the Virtual User (VU) number to represent the Expert User. The default VU number is zero (0).
4. Click OK.

To enable Expert User during the load test:

1. In the Runtime window, click Actions>Set Expert User Options. The Update Expert User Options dialog box displays listing all the scripts that support Expert User counters.
2. Click the scripts in which you want to enable Expert User, then click OK.

Note: Selecting Expert User Scripts at the top level enables or disables expert user on all associated scripts.

Analyzing Load Test Data

Analyzing Load Test Data

By default, load test timing data is sent from the Conductor to Analyze at the end of a load test. Any appropriate server monitoring data is also sent to Analyze and merged into your timing file (.tim).

You can set an option in the Conductor to automatically launch Analyze at the end of a load test (details), or you can open Analyze manually from the Conductor toolbar or your QALoad program group.

Creating a Timing File (.tim)

Once all workstations stop executing, click the Quit Current Test button on the toolbar to complete the test and automatically create the timing file (.tim).

Viewing Test Statistics

Compute and view test statistics in QALoad Analyze.

To access Analyze from the Conductor:

From the Conductor's Tools menu, choose Analyze.

Integration and Server Monitoring

Server and Performance Monitoring

QALoad integrates several mechanisms for merging load test response time data with server utilization data and performance metrics. Select the method that best suits your needs, or for which you are licensed (if applicable). Most methods produce data that is included in your load test timing results and processed in QALoad Analyze. The only exception is Application Vantage. Data captured from ApplicationVantage can be opened in ApplicationVantage, but not in QALoad.

This section briefly describes each method, and provides links to more detailed information about setting up a test that includes the appropriate method.

- Remote Monitoring — allows you to monitor server utilization statistics from a remote machine without installing any software on the remote machine.
- ServerVantage — integrates with your existing ServerVantage installation. You must be licensed for and have installed and configured the appropriate product in order to integrate with QALoad .
- ApplicationVantage — collects test data that you can open in ApplicationVantage.
- Vantage Analyzer - enables you to easily drill into specific problem transactions to determine the cause of bottlenecks in your production applications

Integration and Monitoring Requirements

Integration Requirements

ApplicationVantage

- QALoad 5.7 supports integration with ApplicationVantage 10.0 Service Pack 2 and 10.1.
- Integration with ApplicationVantage is supported on the Windows platform only.

ServerVantage

- QALoad 5.7 supports integration with ServerVantage (SVI Monitoring) 10.0 and 10.1.
- QALoad 5.7 supports integration with ServerVantage (Remote Monitoring) 10.1 Service Pack 1.5 only.

ClientVantage

QALoad supports integration with ClientVantage. QALoad is packaged with the current version of ClientVantage at time of release.

Vantage Analyzer

QALoad 5.7 supports integration with Vantage Analyzer 10.1 Service Pack 1.

Monitoring Requirements

In addition to the integration requirements, your system may need to meet specific requirements to support remote monitoring.

JVM Requirements

Oracle Application Server (AS), JVM, SAP, WebLogic, WebSphere, and WebSphere MQ monitoring all require JVM installed on the Conductor machine.

- For Oracle AS monitoring, if monitoring Oracle AS 10g, you must use Java Virtual Machine 1.5.
- For SAP monitoring, you must use JVM 1.4.
- For WebLogic monitoring version 7 and earlier versions, use JVM 1.3. For WebLogic version 8.1, use JVM 1.4. For WebLogic 9, use JVM 1.5. You may also use the JVM that is distributed with the WebLogic Application Server.
- For WebSphere monitoring, you must use the JVM provided with the WebSphere client or server.
- For WebSphere MQ monitoring, you must use JVM 1.4.

File Installation Requirements

Oracle Application Server (AS), SAP, WebLogic, WebSphere, WebSphere MQ, WMI, and Cold Fusion monitoring require the following files.

Oracle AS

For Oracle AS 10g, you must store copies of the dms.jar, xmlparserv2.jar, ons.jar, and optic.jar files from the monitored Oracle AS server on the Conductor machine.

SAP Monitoring

The SAP files listed below must be placed on the Conductor machine:

- librfc32.dll
- sapjco.jar
- sapjcorfc

To obtain these files, install the SAP Java Connector package (JCo) on the Conductor machine. The JCo package is available from SAP. Add the location of the files, to the Path System Variable of the Conductor machine. For more information, refer to the Requirements for SAP Remote Monitoring topic in the ServerVantage Reconfigure Agent Online Help.

WebLogic Monitoring

The weblogic.jar file must be placed on the Conductor machine. Copy the jar file from the lib directory of the WebLogic application server to a separate directory in the Conductor machine. If you are monitoring WebLogic version 8.1, copy the weblogic.jar and webservices.jar files to the same directory. If you are monitoring WebLogic 9.x, copy the weblogic.jar and wljmxclient.jar file to the same directory. For more information, refer to Requirements for WebLogic Remote Monitoring in the ServerVantage Reconfigure Agent Online Help.

WebSphere Monitoring

Note the following requirements for WebSphere monitoring:

- The WebSphere client files must be installed on the Conductor machine. Installing the WebSphere Application Server Admin Server software on the Conductor machine provides the necessary client files. Note the directory path of the WebSphere\AppServer\Java files. For more information, refer to Requirements for WebSphere Remote Monitoring in the ServerVantage Reconfigure Agent Online Help.
- The Java Home for the monitoring task must be setup for compatible Java version; for example, WebSphere\AppServer\Java.

- If authentication is required and soap.client.props and ssl.client.props authentication files are installed in a custom directory, you must also place copies of the files in WebSphere\AppServer\properties.

WebSphere MQ Monitoring

The WebSphere client files listed below must be placed in a directory on the Conductor machine:

- com.ibm.mq.jar
- com.ibm.mq.pcf.jar
- connector.jar

The files may be obtained from the installation of the WebSphere Application Server Admin Server software on the Conductor machine. If the installation does not include the com.ibm.mq.pcf.jar file, obtain the file from the IBM Support Pac MS0B. See "http://www-1.ibm.com/support/docview.wss?rs=171&uid=swg24000668&loc=en_US&cs=utf-8&lang=en".

For more information, refer to Configuring WebSphere MQ for Remote Monitoring in the ServerVantage Reconfigure Agent Online Help.

WMI Monitoring

WMI security must be enabled on the monitored server machine and the WMI service must be started. For more information, refer to Configuring WMI for Remote Monitoring in the ServerVantage Reconfigure Agent Online Help.

Cold Fusion Monitoring

Performance Monitoring must be enabled from the Cold Fusion Administrator Page – Debugging Settings of the monitored server machine. Cold Fusion is available under Windows Registry monitoring.

Host Verification for QALoad Monitoring

- Ensure host accessibility. Add an entry for the monitored machine to the system hosts file of the Conductor machine. Consult the network administrator for more information.
- Test host availability. Type the following command at the Run command: ping <monitored machine name>.

Remote Monitoring

Overview of Remote Monitoring

Remote Monitoring enables you to extract data from Windows Registry, JVM, Oracle Application Server (AS), SAP, SNMP, ServerVantage, Vantage Analyzer, WebLogic, WebSphere, WebSphere MQ, and WMI counters on the servers under stress without installing any software on the servers.

Note: Select counters for monitor types in the application.

To use Remote Monitoring:

- You must have login access to the machines you want to monitor.
- You must select the servers and counters to monitor on the machines you identify using the monitoring options on Conductor's Manage Monitoring Tasks window.
- To collect SNMP counters, SNMP must be enabled on the Remote Monitor machine. Refer to your operating system help for information about enabling SNMP.
- To collect Windows registry counters, you must have a valid sign-on for the servers under test.

! For requirements for Oracle AS, SAP, WebLogic, WebSphere, WebSphere MQ, and WMI, see Integration and Monitoring Requirements.

QALoad uses the default ports 7790 and 7788 when it communicates with the ServerVantage agent and client. You can override the default ports if your ServerVantage installation requires it.

While your test is running, QALoad collects the appropriate counter data and writes it to your timing file where you can view it in Analyze after the test. What counters are available?

You can simplify the configuration process by creating or applying pre-defined monitoring templates. A monitoring template is a predefined group of counters not associated with a specific machine.

To set up Remote Monitoring, see Creating a New Monitoring Task.

Monitoring Counters

About Counters and Instances

You use counters and, in some cases, specific instances of counters when you monitor servers.

Counters

Counters are the numeric data values that are collected when you monitor servers. Counters exist for components such as processor, memory, processes, hard disk, and cache, with a set of counters that measure statistical information. For Windows, a large number of performance counters are provided by the operating system registry and Windows server applications. Registry counters can monitor external components of the environment such as databases, applications, and printers.

Many of the counters that are collected are points in time data values, such as Process\ thread count. Some counters are cumulative, such as server logon errors, and some are averages, such as the page faults per second in Job Object Details.

In addition to the counters for numeric values, a set of extended data counters is provided for a number of key performance indicators. These extended data counters can provide intelligent data points that have associated textual data for the numeric value. For example, the extended CPU usage counter's intelligent datapoint shows the top 10 processes consuming CPU at that time.

Note: (SNMP) You can use a customized OID file to supplement the standard counters provided by QALoad during counter discovery. See Customizing SNMP Counter Discovery for more information.

Instances

When you select a counter to monitor, the available instances, or occurrences, for that counter appear. Counters can have several instances or no instances. For example, if a system has multiple processors, then the Processor counter has multiple instances. For counters with multiple instances, a list of the available instances for that counter is presented. Many counters also have an instance called _Total, which is an aggregate of the individual instances.

Counters for an object, such as processor, have instances that are numbered, beginning with 0 (zero). A machine with a single processor has an instance of _Total and 0. A dual-processor machine has instances of _Total, 0, and 1. Other instances are based on what is currently running on the server, and the instance list displays these for each process name or service name that is active.

Some instances represent the most recent value for the resource, for example, Processes. This is the number of processes in the computer at the time of data collection. Other instances are average values between the last two measurements.

Windows NT Registry Counters

Windows NT Registry Server Counters

QALoad supports the following MS Windows NT Server counter categories:

Counter Category	Description
Active Server Pages	This object type handles the Active Server Pages device on your system.
Browser	This object type displays Browser Statistics.
Cache	The Cache object type manages memory for rapid access to files. Files on Windows NT are cached in main memory in units of pages. Main memory not being used in the working sets of processes is available to the Cache for this purpose. The Cache preserves file pages in memory for as long as possible to permit access to the data through the file system without accessing the disk.
Context Index	This object type handles the Content Index.
Context Index Filter	This object type handles the Content Index Filter.
ICMP	The ICMP object type includes the counters that describe the rates that ICMP Messages are received and sent by a certain entity using the ICMP protocol. It also describes various error counts for the ICMP protocol.
IP	This object type includes those counters that describe the rates that IP datagrams are received and sent by a certain computer using the IP protocol. It also describes various error counts for the IP protocol.
LogicalDisk	A LogicalDisk object type is a partition on a hard or fixed disk drive and assigned a drive letter, such as C. Disks can be partitioned into distinct sections where they can store file, program, and page data. The disk is read to retrieve these items and written to record changes to them.
Memory	The Memory object type includes those counters that describe the behavior of both real and virtual memory on the computer. Real memory is allocated in units of pages. Virtual memory can exceed real memory in size, causing page traffic as virtual pages are moved between disk and real memory.
Network Interface	The Network Interface Object Type includes those counters that describe the rates that bytes and packets are received and sent over a Network TCP/IP connection. It also describes various error counts for the same connection.
Objects	The Objects object type is a meta-object that contains information about the objects in existence on the computer. This information can be used to detect the unnecessary consumption of computer resources. Each object requires memory to store basic information about the object.
Paging File	This object displays information about the system's Page File(s).
PhysicalDisk	A PhysicalDisk object type is a hard or fixed disk drive. It contains 1 or more logical partitions. Disks are used to store file, program, and paging data. The disk is read to retrieve these items and written to record changes to them.

Process	The Process object type is created when a program is run. All the threads in a process share the same address space and have access to the same data.
Process Address Space	Process Address Space object type displays details about the virtual memory usage and allocation of the selected process.
Processor	The Processor object type includes as instances all processors on the computer. A processor is the part in the computer that performs arithmetic and logical computations, and initiates operations on peripherals. It executes (such as runs) programs on the computer.
Redirector	The Redirector is the object that manages network connections to other computers that originate from your own computer.
Server	The Server object type is the process that interfaces the services from the local computer to the network services.
Server Work Queues	The Server Work Queues object type handles explain text performance data.
SMTP Server	This object type handles the counters specific to the SMTP Server.
System	This object type includes those counters that apply to all processors on the computer collectively. These counters represent the activity of all processors on the computer.
TCP	The TCP object type includes the counters that describe the rates that TCP Segments are received and sent by a certain entity using the TCP protocol. In addition, it describes the number of TCP connections in each possible TCP connection state.
Telephony	This object type handles the Telephony System.
Thread	The Thread object type is the basic object that executes instructions in a processor. Every running process has at least one thread.
UDP	The UDP object type includes the counters that describe the rates that UDP datagrams are received and sent by a certain entity using the UDP protocol. It also describes various error counts for the UDP protocol.

Active Server Pages Counters

QALoad supports the Active Server Pages category for Windows NT. This object type handles these registry counters:

Debugging Requests	Requests Rejected
Errors During Script Runtime	Requests Succeeded
Errors From ASP Preprocessor	Requests Timed Out
Errors From Script Compilers	Requests Total
Errors/Sec	Script Engines Cached

Memory Allocated	Session Duration
Request Bytes In Total	Sessions Current
Request Bytes Out Total	Sessions Timed Out
Request Execution Time	Sessions Total
Request Wait Time	Template Cache Hit Rate
Requests/Sec	Template Notifications
Requests Disconnected	Templates Cached
Requests Executing	Transactions/Sec
Requests Failed Total	Transactions Aborted
Requests Not Authorized	Transactions Committed
Requests Not Found	Transactions Pending
Requests Queued	Transactions Total

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Browser Counters

QALoad supports the Browser category for Windows NT. This object type handles these registry counters:

Announcements Domain/sec	Mailslot Allocations Failed
Announcements Server/sec	Mailslot Opens Failed/sec
Announcements Total/sec	Mailslot Receives Failed
Duplicate Master Announcements	Mailslot Writes/sec
Election Packets/sec	Mailslot Writes Failed
Enumerations Domain/sec	Missed Mailslot Datagrams
Enumerations Other/sec	Missed Server Announcements
Enumerations Server/sec	Missed Server List Requests
Enumerations Total/sec	Server Announce Allocations Failed/sec
Illegal Datagrams/sec	Server List Requests/sec

For information on the registry counters, refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Cache Counters

QALoad supports the Cache category for Windows NT. This object type handles these registry counters:

Async Copy Reads/sec	Fast Reads/sec
Async Data Maps/sec	Lazy Write Flushes/sec
Async Fast Reads/sec	Lazy Write Pages/sec
Async MDL Reads/sec	MDL Read Hits %
Async Pin Reads/sec	MDL Reads/sec
Copy Read Hits %	Pin Read Hits %
Copy Reads/sec	Pin Reads/sec
Data Flush Pages/sec	Read Aheads/sec
Data Flushes/sec	Sync Copy Reads/sec
Data Map Hits %	Sync Data Maps/sec
Data Map Pins/sec	Sync Fast Reads/sec
Data Maps/sec	Sync MDL Reads/sec
Fast Read Not Possibles/sec	Sync Pin Reads/sec
Fast Read Resource Misses/sec	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Content Index Filter Counters

QALoad supports the Content Index Filter category for Windows NT. This object type handles these registry counters:

Binding time (msec)	Total filter speed (MBytes/hr)
Filter speed (MBytes/hr)	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Content Index Counters

QALoad supports the Content Index category for Windows NT. This object type handles these registry counters:

# documents filtered	Running queries
Files to be filtered	Total # documents
Index size (MBytes)	Unique keys
Merge progress	Wordlists
Persistent indexes	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

ICMP Counters

QALoad supports the ICMP category for Windows NT. This object type handles these registry counters:

Messages/sec	Received Timestamp/sec
Messages Outbound Errors	Received Timestamp Reply/sec
Messages Received/sec	Sent Address Mask
Messages Received Errors	Sent Address Mask Reply
Messages Sent/sec	Sent Destination Unreachable
Received Address Mask	Sent Echo/sec
Received Address Mask Reply	Sent Echo Reply/sec
Received Dest. Unreachable	Sent Parameter Problem
Received Echo/sec	Sent Redirect/sec
Received Echo Reply/sec	Sent Source Quench
Received Parameter Problem	Sent Time Exceeded
Received Redirect/sec	Sent Timestamp/sec
Received Source Quench	Sent Timestamp Reply/sec
Received Time Exceeded	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

IP Counters

QALoad supports the IP category for Windows NT. This object type handles these registry counters:

Datagrams/sec	Datagrams Received Unknown Protocol
Datagrams Forwarded/sec	Datagrams Sent/sec
Datagrams Outbound Discarded	Fragment Re-assembly Failures
Datagrams Outbound No Route	Fragmentation Failures
Datagrams Received/sec	Fragmented Datagrams/sec
Datagrams Received Address Errors	Fragments Created/sec
Datagrams Received Delivered/sec	Fragments Re-assembled/sec
Datagrams Received Discarded	Fragments Received/sec
Datagrams Received Header Errors	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

LogicalDisk Counters

QALoad supports the LogicalDisk category for Windows NT. This object type handles these registry counters:

% Disk Read Time	Avg. Disk sec/Write
% Disk Time	Avg. Disk Write Queue Length
% Disk Write Time	Current Disk Queue Length
% Free Space	Disk Bytes/sec
Avg. Disk Bytes/Read	Disk Read Bytes/sec
Avg. Disk Bytes/Transfer	Disk Reads/sec
Avg. Disk Bytes/Write	Disk Transfers/sec
Avg. Disk Queue Length	Disk Write Bytes/sec
Avg. Disk Read Queue Length	Disk Writes/sec
Avg. Disk sec/Read	Free Megabytes
Avg. Disk sec/Transfer	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Memory Counters

QALoad supports the Memory category for Windows NT. This object type handles these registry counters:

% Committed Bytes In Use	Pages Output/sec
Available Bytes	Pool Nonpaged Allocs
Cache Bytes	Pool Nonpaged Bytes
Cache Bytes Peak	Pool Paged Allocs
Cache Faults/sec	Pool Paged Bytes
Commit Limit	Pool Paged Resident Bytes
Committed Bytes	System Cache Resident Bytes
Demand Zero Faults/sec	System Code Resident Bytes
Free System Page Table Entries	System Code Total Bytes
Page Faults/sec	System Driver Resident Bytes
Page Reads/sec	System Driver Total Bytes
Page Writes/sec	Transition Faults/sec
Pages/sec	Write Copies/sec
Pages Input/sec	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Network Interface Counters

QALoad supports the Network Interface category for Windows NT. This object type handles these registry counters:

Bytes Received/sec	Packets Received Discarded
Bytes Sent/sec	Packets Received Errors
Bytes Total/sec	Packets Received Non-Unicast/sec
Current Bandwidth	Packets Received Unicast/sec
Output Queue Length	Packets Received Unknown
Packets/sec	Packets Sent/sec
Packets Outbound Discarded	Packets Sent Non-Unicast/sec
Packets Outbound Errors	Packets Sent Unicast/sec
Packets Received/sec	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Object Counters

QALoad supports the Objects category for Windows NT. This object type handles these registry counters:

Events	Sections
Mutexes	Semaphores
Processes	Threads

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Paging File Counters

QALoad supports the Paging File category for Windows NT. This object type handles these registry counters:

% Usage	% Usage Peak

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Physical Disk Counters

QALoad supports the Physical Disk category for Windows NT. This object type handles these registry counters:

% Disk Read Time	Avg. Disk sec/Write
% Disk Time	Avg. Disk Write Queue Length
% Disk Write Time	Current Disk Queue Length
Avg. Disk Bytes/Read	Disk Bytes/sec
Avg. Disk Bytes/Transfer	Disk Read Bytes/sec
Avg. Disk Bytes/Write	Disk Reads/sec
Avg. Disk Queue Length	Disk Transfers/sec
Avg. Disk Read Queue Length	Disk Write Bytes/sec
Avg. Disk sec/Read	Disk Writes/sec

Avg. Disk sec/Transfer

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Process Address Space Counters

QALoad supports the Process Address category for Windows NT. This object type handles these registry counters:

Bytes Free	Mapped Space Read Only
Bytes Image Free	Mapped Space Write Copy
Bytes Image Reserved	Reserved Space Exec Read/Write
Bytes Reserved	Reserved Space Exec Read Only
ID Process	Reserved Space Exec Write Copy
Image Space Exec Read/Write	Reserved Space Executable
Image Space Exec Read Only	Reserved Space No Access
Image Space Exec Write Copy	Reserved Space Read/Write
Image Space Executable	Reserved Space Read Only
Image Space No Access	Reserved Space Write Copy
Image Space Read/Write	Unassigned Space Exec Read/Write
Image Space Read Only	Unassigned Space Exec Read Only
Image Space Write Copy	Unassigned Space Exec Write Copy
Mapped Space Exec Read/Write	Unassigned Space Executable
Mapped Space Exec Read Only	Unassigned Space No Access
Mapped Space Exec Write Copy	Unassigned Space Read/Write
Mapped Space Executable	Unassigned Space Read Only
Mapped Space No Access	Unassigned Space Write Copy
Mapped Space Read/Write	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Process Counters

QALoad supports the Process category for Windows NT. This object type handles these registry counters:

% Privileged Time	Pool Nonpaged Bytes
% Processor Time (See Note below)	Pool Paged Bytes
% User Time	Priority Base
Elapsed Time	Private Bytes
Handle Count	Thread Count
ID Process	Virtual Bytes
Page Faults/sec	Virtual Bytes Peak
Page File Bytes	Working Set
Page File Bytes Peak	Working Set Peak

Note: If you use the % Processor Time counter in an event rule, set the event rule to trigger after two or more occurrences of the event. The CPU consumption for the first datapoint sample is artificially high because the agent is starting the task.

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Processor Counters

QALoad supports the Processor category for Windows NT. This object type handles these registry counters:

% DPC Time	APC Bypasses/sec
% Interrupt Time	DPC Bypasses/sec
% Privileged Time	DPC Rate
% Processor Time	DPCs Queued/sec
% User Time	Interrupts/sec

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Redirector Counters

QALoad supports the Redirector category for Windows NT. This object type handles these registry counters:

Bytes Received/sec	Read Operations Random/sec
Bytes Total/sec	Read Packets/sec

Bytes Transmitted/sec	Read Packets Small/sec
Connects Core	Reads Denied/sec
Connects Lan Manager 2.0	Reads Large/sec
Connects Lan Manager 2.1	Server Disconnects
Connects Windows NT	Server Reconnects
Current Commands	Server Sessions
File Data Operations/sec	Server Sessions Hung
File Read Operations/sec	Write Bytes Cache/sec
File Write Operations/sec	Write Bytes Network/sec
Network Errors/sec	Write Bytes Non-Paging/sec
Packets/sec	Write Bytes Paging/sec
Packets Received/sec	Write Operations Random/sec
Packets Transmitted/sec	Write Packets/sec
Read Bytes Cache/sec	Write Packets Small/sec
Read Bytes Network/sec	Writes Denied/sec
Read Bytes Non-Paging/sec	Writes Large/sec
Read Bytes Paging/sec	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Server Counters

QALoad supports the Server category for Windows NT. This object type handles these registry counters:

Blocking Requests Rejected	Logon Total
Bytes Received/sec	Pool Nonpaged Bytes
Bytes Total/sec	Pool Nonpaged Failures
Bytes Transmitted/sec	Pool Nonpaged Peak
Context Blocks Queued/sec	Pool Paged Bytes
Errors Access Permissions	Pool Paged Failures
Errors Granted Access	Pool Paged Peak

Errors Logon	Server Sessions
Errors System	Sessions Errored Out
File Directory Searches	Sessions Forced Off
Files Open	Sessions Logged Off
Files Opened Total	Sessions Timed Out
Logon/sec	Work Item Shortages

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Server Work Queues Counters

QALoad supports the Server Work Queues category for Windows NT. This object type handles these registry counters:

Active Threads	Queue Length
Available Threads	Read Bytes/sec
Available Work Items	Read Operations/sec
Borrowed Work Items	Total Bytes/sec
Bytes Received/sec	Total Operations/sec
Bytes Sent/sec	Work Item Shortages
Bytes Transferred/sec	Write Bytes/sec
Context Blocks Queued/sec	Write Operations/sec
Current Clients	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

SMTP Server Counters

QALoad supports the SMTP Server category for Windows NT. This object type handles these registry counters:

% Recipients Local	Message Bytes Received/sec
% Recipients Remote	Message Bytes Received Total
Avg Recipients/msg Received	Message Bytes Sent/sec

Avg Recipients/msg Sent	Message Bytes Sent Total
Avg Retries/msg Delivered	Message Bytes Total
Avg Retries/msg Sent	Message Bytes Total/sec
Base % Recipients Local	Message Delivery Retries
Base % Recipients Remote	Message Received/sec
Base Avg Recipients/msg Received	Message Send Retries
Base Avg Recipients/msg Sent	Messages Delivered/sec
Base Avg Retries/msg Delivered	Messages Delivered Total
Base Avg Retries/msg Sent	Messages Received Total
Bytes Received/sec	Messages Refused for Address Objects
Bytes Received Total	Messages Refused for Mail Objects
Bytes Sent/sec	Messages Refused for Size
Bytes Sent Total	Messages Retrieved/sec
Bytes Total	Messages Retrieved Total
Bytes Total/sec	Messages Sent/sec
Connection Errors/sec	Messages Sent Total
Directory Drops/sec	NDRs Generated
Directory Drops Total	Number of MailFiles Open
Directory Pickup Queue Length	Number of QueueFiles Open
DNS Queries/sec	Outbound Connections Current
DNS Queries Total	Outbound Connections Refused
ETRN Messages/sec	Outbound Connections Total
ETRN Messages Total	Remote Queue Length
Inbound Connections Current	Remote Retry Queue Length
Inbound Connections Total	Routing Table Lookups/sec
Local Queue Length	Routing Table Lookups Total
Local Retry Queue Length	Total Connection Errors

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

System Counters

QALoad supports the System category for Windows NT. This object type handles these registry counters:

% Registry Quota In Use	File Read Operations/sec
% Total DPC Time	File Write Bytes/sec
% Total Interrupt Time	File Write Operations/sec
% Total Privileged Time	Floating Emulations/sec
% Total Processor Time	Processor Queue Length
% Total User Time	System Calls/sec
Alignment Fixups/sec	System Up Time
Context Switches/sec	Total APC Bypasses/sec
Exception Dispatches/sec	Total DPC Bypasses/sec
File Control Bytes/sec	Total DPC Rate
File Control Operations/sec	Total DPCs Queued/sec
File Data Operations/sec	Total Interrupts/sec
File Read Bytes/sec	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

TCP Counters

QALoad supports the TCP category for Windows NT. This object type handles these registry counters:

Connection Failures	Segments/sec
Connections Active	Segments Received/sec
Connections Established	Segments Retransmitted/sec
Connections Passive	Segments Sent/sec
Connections Reset	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Telephony Counters

QALoad supports the Telephony category for Windows NT. This object type handles these registry counters:

Active Lines	Incoming Calls/sec
Active Telephones	Lines
Client Apps	Outgoing Calls/sec
Current Incoming Calls	Telephone Devices
Current Outgoing Calls	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Thread Counters

QALoad supports the Thread category for Windows NT. This object type handles these registry counters:

% Privileged Time	ID Thread
% Processor Time	Priority Base
% User Time	Priority Current
Context Switches/sec	Start Address
Elapsed Time	Thread State
ID Process	Thread Wait Reason

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

UDP Counters

QALoad supports the UDP category for Windows NT. This object type handles these registry counters:

Datagrams/sec	Datagrams Received Errors
Datagrams No Port/sec	Datagrams Sent/sec
Datagrams Received/sec	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Windows Win2K Registry Counters

Windows Win2K Server Registry Counters

Remote Monitoring Agents can monitor the same Windows registry counters as PERFMON, the performance monitoring application available with the Windows operating system. The Windows registry option monitors machines that run Windows 2000 and XP. To retrieve Windows Registry Counters, you must have access, via a user name and password, to the remote machine.

QALoad supports the following MS Windows counter categories:

Counter Category	Description
ACS/RSVP Service	RSVP or ACS service performance counters.
Active Server Pages	This object type handles the Active Server Pages device on your system.
Browser	The Browser performance object consists of counters that measure the rates of announcements, enumerations, and other Browser transmissions.
Cache	The Cache performance object consists of counters that monitor the file system cache, an area of physical memory that stores recently used data as long as possible to permit access to the data without reading from the disk. Because applications typically use the cache, the cache is monitored as an indicator of application I/O operations. When memory is plentiful, the cache can grow, but when memory is scarce, the cache can become too small to be effective.
IAS Accounting Clients	IAS Accounting Clients
IAS Accounting Server	IAS Accounting Server
IAS Authentication Clients	IAS Authentication Clients
IAS Authentication Server	IAS Authentication Server
ICMP	The ICMP performance object consists of counters that measure the rates at which messages are sent and received by using ICMP protocols. It also includes counters that monitor ICMP protocol errors.
IP	The IP performance object consists of counters that measure the rates at which IP datagrams are sent and received by using IP protocols. It also includes counters that monitor IP protocol errors.
LogicalDisk	The Logical Disk performance object consists of counters that monitor logical partitions of a hard or fixed disk drives. Performance Monitor identifies logical disks by their a drive letter, such as C.
Memory	The Memory performance object consists of counters that describe the behavior of physical and virtual memory on the computer. Physical memory is the amount of random access memory on the computer.. Virtual memory consists of the space in physical memory and on disk. Many of the memory counters monitor paging, which is the movement of pages of code and data between disk and physical memory. Excessive paging, a symptom of a memory shortage, can cause delays which interfere with all system processes.
NBT Connection	The NBT Connection performance object consists of counters that measure the rates at which bytes are sent and received over the NBT

	connection between the local computer and a remote computer. The connection is identified by the name of the remote computer.
Network Interface	The Network Interface performance object consists of counters that measure the rates at which bytes and packets are sent and received over a TCP/IP network connection. It includes counters that monitor connection errors.
Objects	The Object performance object consists of counters that monitor logical objects in the system, such as processes, threads, mutexes, and semaphores. This information can be used to detect the unnecessary consumption of computer resources. Each object requires memory to store basic information about the object.
Paging File	The Paging File performance object consists of counters that monitor the paging file(s) on the computer. The paging file is a reserved space on disk that backs up committed physical memory on the computer.
PhysicalDisk	The Physical Disk performance object consists of counters that monitor hard or fixed disk drive on a computer. Disks are used to store file, program, and paging data and are read to retrieve these items, and written to record changes to them. The values of physical disk counters are sums of the values of the logical disks (or partitions) into which they are divided.
Print Queue	Displays performance statistics about a Print Queue.
Process	The Process performance object consists of counters that monitor running application program and system processes. All the threads in a process share the same address space and have access to the same data.
Process Address Space	The Process Address Space performance object consists of counters that monitor memory allocation and use for a selected process.
Processor	The Processor performance object consists of counters that measure aspects of processor activity The processor is the part of the computer that performs arithmetic and logical computations, initiates operations on peripherals, and runs the threads of processes. A computer can have multiple processors. The processor object represents each processor as an instance of the object.
Redirector	The Redirector performance object consists of counter that monitor network connections originating at the local computer.
Server	The Server performance object consists of counters that measure communication between the local computer and the network.
Server Work Queues	The Server Work Queues performance object consists of counters that monitor the length of the queues and objects in the queues.
SMTP NTFS Store Driver	This object represents global counters for the Exchange NTFS Store driver.
SMTP Server	The counters specific to the SMTP Server.
System	The System performance object consists of counters that apply to more than one instance of a component processors on the computer.

TCP	The TCP performance object consists of counters that measure the rates at which TCP Segments are sent and received by using the TCP protocol. It includes counters that monitor the number of TCP connections in each TCP connection state.
Telephony	The Telephony System.
Thread	The Thread performance object consists of counters that measure aspects of thread behavior. A thread is the basic object that executes instructions on a processor. All running processes have at least one thread.
UDP	The UDP performance object consists of counters that measure the rates at which UDP datagrams are sent and received by using the UDP protocol. It includes counters that monitor UDP protocol errors.

ACS RSVP Service Counters

QALoad supports the ACS RSVP Service category for Windows. This object type handles these registry counters:

API notifications	Interfaces
API sockets	Network sockets
Bytes in API notifies	PATH from API
Failed API requests	RESV from API
Failed API sends	RSVP msg buffers in use
GQOS sessions	Timers

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Active Server Pages Counters

QALoad supports the Active Server Pages category for Windows. This object type handles these registry counters:

Debugging Requests	Requests Succeeded
Errors/Sec	Requests Timed Out
Errors During Script Runtime	Requests Total
Errors From ASP Preprocessor	Script Engines Cached
Errors From Script Compilers	Session Duration
Request Bytes In Total	Sessions Current

Request Bytes Out Total	Sessions Timed Out
Request Execution Time	Sessions Total
Request Wait Time	Template Cache Hit Rate
Requests/Sec	Template Notifications
Requests Disconnected	Templates Cached
Requests Executing	Transactions/Sec
Requests Failed Total	Transactions Aborted
Requests Not Authorized	Transactions Committed
Requests Not Found	Transactions Pending
Requests Queued	Transactions Total
Requests Rejected	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Browser Counters

QALoad supports the Browser category for Windows. This object type handles these registry counters:

Announcements Domain/sec	Mailslot Allocations Failed
Announcements Server/sec	Mailslot Opens Failed/sec
Announcements Total/sec	Mailslot Receives Failed
Duplicate Master Announcements	Mailslot Writes/sec
Election Packets/sec	Mailslot Writes Failed
Enumerations Domain/sec	Missed Mailslot Datagrams
Enumerations Other/sec	Missed Server Announcements
Enumerations Server/sec	Missed Server List Requests
Enumerations Total/sec	Server Announce Allocations Failed/sec
Illegal Datagrams/sec	Server List Requests/sec

For information on the registry counters, refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Cache Win2K Counters

QALoad supports the Cache category for Windows. This object type handles these registry counters:

Async Copy Reads/sec	Fast Reads/sec
Async Data Maps/sec	Lazy Write Flushes/sec
Async Fast Reads/sec	Lazy Write Pages/sec
Async MDL Reads/sec	MDL Read Hits %
Async Pin Reads/sec	MDL Reads/sec
Copy Read Hits %	Pin Read Hits %
Copy Reads/sec	Pin Reads/sec
Data Flush Pages/sec	Read Aheads/sec
Data Flushes/sec	Sync Copy Reads/sec
Data Map Hits %	Sync Data Maps/sec
Data Map Pins/sec	Sync Fast Reads/sec
Data Maps/sec	Sync MDL Reads/sec
Fast Read Not Possibles/sec	Sync Pin Reads/sec
Fast Read Resource Misses/sec	

For information on the registry counters, refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

IAS Accounting Clients Counters

QALoad supports the IAS Accounting Clients category for Windows. This object type handles these registry counters:

Accounting-Requests	Malformed Packets
Accounting-Requests/sec	Malformed Packets/sec
Accounting-Responses	No Record
Accounting-Responses/sec	No Record/sec
Bad Authenticators	Packets Received
Bad Authenticators/sec	Packets Received/sec
Dropped Packets	Packets Sent

Dropped Packets/sec	Packets Sent/sec
Duplicate Accounting-Requests	Unknown Type
Duplicate Accounting-Requests/sec	Unknown Type/sec

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

IAS Accounting Server Counters

QALoad supports the IAS Accounting Server category for Windows. This object type handles these registry counters:

Accounting-Requests	Malformed Packets
Accounting-Requests/sec	Malformed Packets/sec
Accounting-Responses	No Record
Accounting-Responses/sec	No Record/sec
Bad Authenticators	Packets Received
Bad Authenticators/sec	Packets Received/sec
Dropped Packets	Packets Sent
Dropped Packets/sec	Packets Sent/sec
Duplicate Accounting-Requests	Server Reset Time
Duplicate Accounting-Requests/sec	Server Up Time
Invalid Requests	Unknown Type
Invalid Requests/sec	Unknown Type/sec

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

IAS Authentication Clients Win2K Counters

QALoad supports the IAS Authentication Clients category for Windows. This object type handles these registry counters:

Access-Accepts	Dropped Packets/sec
Access-Accepts/sec	Duplicate Access-Requests
Access-Challenges	Duplicate Access-Requests/sec

Access-Challenges/sec	Malformed Packets
Access-Rejects	Malformed Packets/sec
Access-Rejects/sec	Packets Received
Access-Requests	Packets Received/sec
Access-Requests/sec	Packets Sent
Bad Authenticators	Packets Sent/sec
Bad Authenticators/sec	Unknown Type
Dropped Packets	Unknown Type/sec

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

IAS Authentication Server Counters

QALoad supports the IAS Authentication Server category for Windows. This object type handles these registry counters:

Access-Accepts	Duplicate Access-Requests/sec
Access-Accepts/sec	Invalid Requests
Access-Challenges	Invalid Requests/sec
Access-Challenges/sec	Malformed Packets
Access-Rejects	Malformed Packets/sec
Access-Rejects/sec	Packets Received
Access-Requests	Packets Received/sec
Access-Requests/sec	Packets Sent
Bad Authenticators	Packets Sent/sec
Bad Authenticators/sec	Server Reset Time
Dropped Packets	Server Up Time
Dropped Packets/sec	Unknown Type
Duplicate Access-Requests	Unknown Type/sec

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

ICMP Counters

QALoad supports the ICMP category for Windows. This object type handles these registry counters:

Messages/sec	Received Timestamp/sec
Messages Outbound Errors	Received Timestamp Reply/sec
Messages Received/sec	Sent Address Mask
Messages Received Errors	Sent Address Mask Reply
Messages Sent/sec	Sent Destination Unreachable
Received Address Mask	Sent Echo/sec
Received Address Mask Reply	Sent Echo Reply/sec
Received Dest. Unreachable	Sent Parameter Problem
Received Echo/sec	Sent Redirect/sec
Received Echo Reply/sec	Sent Source Quench
Received Parameter Problem	Sent Time Exceeded
Received Redirect/sec	Sent Timestamp/sec
Received Source Quench	Sent Timestamp Reply/sec
Received Time Exceeded	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

IP Counters

QALoad supports the IP category for Windows. This object type handles these registry counters:

Datagrams/sec	Datagrams Received Unknown Protocol
Datagrams Forwarded/sec	Datagrams Sent/sec
Datagrams Outbound Discarded	Fragment Re-assembly Failures
Datagrams Outbound No Route	Fragmentation Failures
Datagrams Received/sec	Fragmented Datagrams/sec
Datagrams Received Address Errors	Fragments Created/sec
Datagrams Received Delivered/sec	Fragments Re-assembled/sec
Datagrams Received Discarded	Fragments Received/sec

Datagrams Received Header Errors

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

LogicalDisk Counters

QALoad supports the LogicalDisk category for Windows. This object type handles these registry counters:

% Disk Read Time	Avg. Disk sec/Write
% Disk Time	Avg. Disk Write Queue Length
% Disk Write Time	Current Disk Queue Length
% Free Space	Disk Bytes/sec
% Idle Time	Disk Read Bytes/sec
Avg. Disk Bytes/Read	Disk Reads/sec
Avg. Disk Bytes/Transfer	Disk Transfers/sec
Avg. Disk Bytes/Write	Disk Write Bytes/sec
Avg. Disk Queue Length	Disk Writes/sec
Avg. Disk Read Queue Length	Free Megabytes
Avg. Disk sec/Read	Split IO/Sec
Avg. Disk sec/Transfer	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Memory Counters

QALoad supports the Memory category for Windows. This object type handles these registry counters:

% Committed Bytes In Use	Pool Nonpaged Allocs
Available Bytes	Pool Nonpaged Bytes
Cache Bytes	Pool Paged Allocs
Cache Bytes Peak	Pool Paged Bytes
Cache Faults/sec	Pool Paged Resident Bytes

Commit Limit	System Cache Resident Bytes
Committed Bytes	System Code Resident Bytes
Demand Zero Faults/sec	System Code Total Bytes
Free System Page Table Entries	System Driver Resident Bytes
Page Faults/sec	System Driver Total Bytes
Page Reads/sec	System VLM Commit Charge
Page Writes/sec	System VLM Commit Charge Peak
Pages/sec	System VLM Shared Commit Charge
Pages Input/sec	Transition Faults/sec
Pages Output/sec	Write Copies/sec

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

NBT Connection Counters

QALoad supports the NBT Connection category for Windows. This object type handles these registry counters:

Bytes Received/sec	Bytes Total/sec
Bytes Sent/sec	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Network Interface Counters

QALoad supports the Network Interface category for Windows. This object type handles these registry counters:

Bytes Received/sec	Packets Received Discarded
Bytes Sent/sec	Packets Received Errors
Bytes Total/sec	Packets Received Non-Unicast/sec
Current Bandwidth	Packets Received Unicast/sec
Output Queue Length	Packets Received Unknown
Packets/sec	Packets Sent/sec

Packets Outbound Discarded	Packets Sent Non-Unicast/sec
Packets Outbound Errors	Packets Sent Unicast/sec
Packets Received/sec	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Objects Counters

QALoad supports the Objects category for Windows. This object type handles these registry counters:

Events	Sections
Mutexes	Semaphores
Processes	Threads

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Paging File Counters

QALoad supports the Paging File category for Windows. This object type handles these registry counters:

% Usage	% Usage Peak

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

PhysicalDisk Counters

QALoad supports the PhysicalDisk category for Windows. This object type handles these registry counters:

% Disk Read Time	Avg. Disk sec/Write
% Disk Time	Avg. Disk Write Queue Length
% Disk Write Time	Current Disk Queue Length
% Idle Time	Disk Bytes/sec
Avg. Disk Bytes/Read	Disk Read Bytes/sec
Avg. Disk Bytes/Transfer	Disk Reads/sec
Avg. Disk Bytes/Write	Disk Transfers/sec

Avg. Disk Queue Length	Disk Write Bytes/sec
Avg. Disk Read Queue Length	Disk Writes/sec
Avg. Disk sec/Read	Split IO/Sec
Avg. Disk sec/Transfer	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Print Queue Counters

QALoad supports the Print Queue category for Windows. This object type handles these registry counters:

Add Network Printer Calls	Max References
Bytes Printed/sec	Not Ready Errors
Enumerate Network Printer Calls	Out of Paper Errors
Job Errors	References
Jobs	Total Jobs Printed
Jobs Spooling	Total Pages Printed
Max Jobs Spooling	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Process Address Space Counters

QALoad supports the Process Address Space category for Windows. This object type handles these registry counters:

Bytes Free	Mapped Space Read Only
Bytes Image Free	Mapped Space Write Copy
Bytes Image Reserved	Reserved Space Exec Read/Write
Bytes Reserved	Reserved Space Exec Read Only
ID Process	Reserved Space Exec Write Copy
Image Space Exec Read/Write	Reserved Space Executable
Image Space Exec Read Only	Reserved Space No Access
Image Space Exec Write Copy	Reserved Space Read/Write

Image Space Executable	Reserved Space Read Only
Image Space No Access	Reserved Space Write Copy
Image Space Read/Write	Unassigned Space Exec Read/Write
Image Space Read Only	Unassigned Space Exec Read Only
Image Space Write Copy	Unassigned Space Exec Write Copy
Mapped Space Exec Read/Write	Unassigned Space Executable
Mapped Space Exec Read Only	Unassigned Space No Access
Mapped Space Exec Write Copy	Unassigned Space Read/Write
Mapped Space Executable	Unassigned Space Read Only
Mapped Space No Access	Unassigned Space Write Copy
Mapped Space Read/Write	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Process Counters

QALoad supports the Process category for Windows. This object type handles these registry counters:

% Privileged Time	IO Write Operations/sec
% Processor Time (See Note below.)	Page Faults/sec
% User Time	Page File Bytes
Creating Process ID	Page File Bytes Peak
Elapsed Time	Pool Nonpaged Bytes
Handle Count	Pool Paged Bytes
ID Process	Priority Base
IO Data Bytes/sec	Private Bytes
IO Data Operations/sec	Thread Count
IO Other Bytes/sec	Virtual Bytes
IO Other Operations/sec	Virtual Bytes Peak
IO Read Bytes/sec	Working Set
IO Read Operations/sec	Working Set Peak

IO Write Bytes/sec

Note: If you use the % Processor Time counter in an event rule, set the event rule to trigger after two or more occurrences of the event. The CPU consumption for the first datapoint sample is artificially high because the agent is starting the task.

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Processor Counters

QALoad supports the Processor category for Windows. This object type handles these registry counters:

% DPC Time	APC Bypasses/sec
% Interrupt Time	DPC Bypasses/sec
% Privileged Time	DPC Rate
% Processor Time	DPCs Queued/sec
% User Time	Interrupts/sec

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Redirector Counters

QALoad supports the Redirector category for Windows. This object type handles these registry counters:

Bytes Received/sec	Read Operations Random/sec
Bytes Total/sec	Read Packets/sec
Bytes Transmitted/sec	Read Packets Small/sec
Connects Core	Reads Denied/sec
Connects Lan Manager 2.0	Reads Large/sec
Connects Lan Manager 2.1	Server Disconnects
Connects Windows NT	Server Reconnects
Current Commands	Server Sessions
File Data Operations/sec	Server Sessions Hung
File Read Operations/sec	Write Bytes Cache/sec
File Write Operations/sec	Write Bytes Network/sec

Network Errors/sec	Write Bytes Non-Paging/sec
Packets/sec	Write Bytes Paging/sec
Packets Received/sec	Write Operations Random/sec
Packets Transmitted/sec	Write Packets/sec
Read Bytes Cache/sec	Write Packets Small/sec
Read Bytes Network/sec	Writes Denied/sec
Read Bytes Non-Paging/sec	Writes Large/sec
Read Bytes Paging/sec	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Server Counters

QALoad supports the Server category for Windows. This object type handles these registry counters:

Blocking Requests Rejected	Logon Total
Bytes Received/sec	Pool Nonpaged Bytes
Bytes Total/sec	Pool Nonpaged Failures
Bytes Transmitted/sec	Pool Nonpaged Peak
Context Blocks Queued/sec	Pool Paged Bytes
Errors Access Permissions	Pool Paged Failures
Errors Granted Access	Pool Paged Peak
Errors Logon	Server Sessions
Errors System	Sessions Errored Out
File Directory Searches	Sessions Forced Off
Files Open	Sessions Logged Off
Files Opened Total	Sessions Timed Out
Logon/sec	Work Item Shortages

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Server Work Queues Counters

QALoad supports the Server Work Queues category for Windows. This object type handles these registry counters:

Active Threads	Queue Length
Available Threads	Read Bytes/sec
Available Work Items	Read Operations/sec
Borrowed Work Items	Total Bytes/sec
Bytes Received/sec	Total Operations/sec
Bytes Sent/sec	Work Item Shortages
Bytes Transferred/sec	Write Bytes/sec
Context Blocks Queued/sec	Write Operations/sec
Current Clients	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

SMTP NTFS Store Drive Counters

QALoad supports the SMTP NTFS Store Drive category for Windows. This object type handles these registry counters:

Messages allocated	Messages in the queue directory
Messages deleted	Open message bodies
Messages enumerated	Open message streams

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

SMTP Server Counters

QALoad supports the SMTP Server category for Windows. This object type handles these registry counters:

% Recipients Local	Local Retry Queue Length
% Recipients Remote	Message Bytes Received/sec
Avg Recipients/msg Received	Message Bytes Received Total
Avg Recipients/msg Sent	Message Bytes Sent/sec
Avg Retries/msg Delivered	Message Bytes Sent Total

Avg Retries/msg Sent

Badmailed Messages (Bad Pickup File)

Badmailed Messages (General Failure)

Badmailed Messages (Hop Count Exceeded)

Badmailed Messages (NDR of DSN)

Badmailed Messages (No Recipients)

Badmailed Messages (Triggered via Event)

Base % Recipients Local

Base % Recipients Remote

Base Avg Recipients/msg Received

Base Avg Recipients/msg Sent

Base Avg Retries/msg Delivered

Base Avg Retries/msg Sent

Bytes Received/sec

Bytes Received Total

Bytes Sent Total

Bytes Sent/sec

Bytes Total

Bytes Total/sec

Categorizer Queue Length

Connection Errors/sec

Current Messages in Local Delivery

Directory Drops/sec

Directory Drops Total

DNS Queries/sec

DNS Queries Total

ETRN Messages/sec

ETRN Messages Total

Inbound Connections Current

Message Bytes Total

Message Bytes Total/sec

Message Delivery Retries

Message Received/sec

Message Send Retries

Messages Currently Undeliverable

Messages Delivered/sec

Messages Delivered Total

Messages Pending Routing

Messages Received Total

Messages Refused for Address Objects

Messages Refused for Mail Objects

Messages Refused for Size

Messages Sent/sec

Messages Sent Total

NDRs Generated

Number of MailFiles Open

Number of QueueFiles Open

Outbound Connections Current

Outbound Connections Refused

Outbound Connections Total

Pickup Directory Messages Retrieved/sec

Pickup Directory Messages Retrieved Total

Remote Queue Length

Remote Retry Queue Length

Routing Table Lookups/sec

Routing Table Lookups Total

Total Connection Errors

Total DSN Failures

Inbound Connections Total	Total messages submitted
Local Queue Length	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

System Counters

QALoad supports the System category for Windows. This object type handles these registry counters:

% Registry Quota In Use	File Write Bytes/sec
Alignment Fixups/sec	File Write Operations/sec
Context Switches/sec	Floating Emulations/sec
Exception Dispatches/sec	Processes
File Control Bytes/sec	Processor Queue Length
File Control Operations/sec	System Calls/sec
File Data Operations/sec	System Up Time
File Read Bytes/sec	Threads
File Read Operations/sec	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

TCP Counters

QALoad supports the TCP category for Windows. This object type handles these registry counters:

Connection Failures	Segments/sec
Connections Active	Segments Received/sec
Connections Established	Segments Retransmitted/sec
Connections Passive	Segments Sent/sec
Connections Reset	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Telephony Counters

QALoad supports the Telephony category for Windows. This object type handles these registry counters:

Active Lines	Incoming Calls/sec
Active Telephones	Lines
Client Apps	Outgoing Calls/sec
Current Incoming Calls	Telephone Devices
Current Outgoing Calls	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

Thread Counters

QALoad supports the Thread category for Windows. This object type handles this registry counter:

User PC

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

UCP Counters

QALoad supports the UCP category for Windows. This object type handles these registry counters:

Datagrams/sec	Datagrams Received Errors
Datagrams No Port/sec	Datagrams Sent/sec
Datagrams Received/sec	

For information on the registry counters refer to the documentation or developer network for that product or the developer kit provided with the product. For Microsoft products, refer to http://msdn.microsoft.com/library/default.asp.

SAP Counters

SAP R/3 Remote Extended Counters

The following extended SAP R/3 remote counters are provided. These counters extend the monitoring of your SAP R/3 system:

Active Servers	Page/Roll Area
Active Users	Page/Roll Area Max
Alerts	Process Monitoring
Buffer Statistics	Spool Queue
CCMS Monitoring	System Log Entries

Connection Test (SM59)

CPU Consumption

Itemized Active Users

Itemized Job Status

Itemized Spool Queue

Job Status

Memory Usage

Number of Dumps

Top CPU Utilization

Top Load

User Function Call

Workload Statistic

Work Processes

SAP Active Servers

This counter returns the active SAP application servers for a specified SAP R/3 instance.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Server Count

Maximum number of servers on which to report data. The default is **10**. The value can range between **00** and **100**.

*D*ata Point

Primary Data Point

The primary data point (PDP) is the number of active SAP application servers in the specified SAP instance. If an error is encountered during data collection, the counter returns 999.

Intelligent Data Point

The intelligent data point (IDP) displays the following information for each server:

Name	Full application server name.
Hostname	Name of application server host.
Type	Service name.
IP	Application server host IP address.
Num Services	Service port number.

Interval

Recommended minimum is 5 minutes.

SAP Active Users

This counter returns all SAP users connected to either a specific SAP R/3 instance or system-wide.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. Enter an instance name as a string in the format:

```
<System name>-<Application server name>-<R/3 system number>-<Client number>
```

For example:

```
C11-sapappserver-01-001
```

SAP User Count

Maximum number of servers on which to report data. The default is **`10`**. The value can range between **`00`** and **`100`**.

Level

The monitoring level. This parameter is pre-defined and single-selectable. Possible values are:

Selected instance only (default)	Only users in the instance specified by the SAP Instance parameter are reported.
All instances in the system	All users of any instance available through the specified instance are reported.

Data Point

Primary Data Point

The primary data point (PDP) is the current queue depth as a percentage of the defined maximum. The Level parameter impacts the number of servers that will be scanned. If an error is encountered during data collection, the counter returns 999.

Intelligent Data Point

The intelligent data point (IDP) displays the following information:

Sysname	Full application server name.
TerminalID	Terminal identification.
Client	User's logon client number.
Username	Name of the user.
Report/Tcode	Name of tcode or report currently used by user.
Terminal	Terminal name.
Time	Dialog time.
Sessions	Number of user sessions.

Interval

Recommended minimum is 5 minutes.

SAP Alerts

This counter returns a description of all the SAP alerts for the specified severity level.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Monitor Set

Name of the monitor set. You can specify one or more sets. In any combination, select values from the discovered list, or enter values manually.

Monitor

Name of the monitor within the selected monitor set. You can specify one or more monitors. In any combination, select values from the discovered list, or enter values manually.

Severity

Alert severity level you want to monitor. This parameter is pre-defined and multi-selectable. Possible values are:

Error - Red (default)

Warning - Yellow

Pattern

Pattern to search for in result. The default is all (* wildcard). You can either accept the default or enter a string. Wildcard characters cannot be included in the string.

Show Alert Text

Specify whether to show the alert's text. This parameter is pre-defined and single-selectable. Possible values are:

Yes

No (default)

Alert Type

Select whether data returned presents only current alert activity or presents a history of activity. This parameter is pre-defined and multi-selectable. Possible values are:

Active alerts (default)

Alert history

Show last minutes

Number of minutes of data history to return.

Data Point

Primary Data Point

A primary data point (PDP) is returned for each combination of parameters. The value returned is the number of alerts of the specified type. If an error is encountered during data collection, the counter returns 999.

Intelligent Data Point

The intelligent data point (IDP) displays the following information:

Color	Red for errors and yellow for warnings
Severity	Severity of alert
Date/Time	Alert timestamp
Alert Unique ID	Alert ID
Status	Status (for example: active, cone, auto completed)
System	MTE system name
Context	MTE context
Object	MTE object
Short Name	MTE short name
Alert Text	Alert text, if any.

Interval

Recommended minimum is 5 minutes.

SAP Buffer Statistic

This counter returns statistics for the specified SAP R/3 buffers.

The primary data point returns the buffer hit ratio, which is an indicator of how efficiently the buffer is being used. For a frequently accessed buffer, the hit ratio should exceed 95%.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Buffer Name

Name of the SAP R/3 buffer you want to monitor. You can specify one or more buffers. In any combination, select values from the discovered list, or enter values manually.

Statistic Name

The SAP R/3 buffer statistic to be used for the primary data point. This parameter is predefined and single-selectable. Possible values are:

% of active objects

% of free objects

Free Size (%)

Free Size (KB)

Hit rate SAP buffer (%) (default)

Maximum no. of objects

No. of active objects

No. of database accesses

No. of free objects

No. of objects swapped

Size of allocated address space (KB)

Storage space available (KB)

Used size (%)

Used size (KB)

Data Point

Primary Data Point

A primary data point (PDP) is returned for each combination of parameters. The primary data point is the value for the statistic specified in the Statistic Name parameter. If an error is encountered during data collection, the counter returns 999.

Intelligent Data Point

The intelligent data point (IDP) lists the values returned for all statistics.

Interval

Recommended minimum is 5 minutes.

SAP CCMS Monitoring

The Computer Center Management System (CCMS) Monitoring counter returns the value of the R/3 CCMS Monitoring Tree Element (MTE) as in R/3 transaction RZ20. The performance, status and log attributes are distinguished. Each MTE in CCMS is represented using four elements: system name, context, object and name. For example, CW2\ Database\ Tablespaces\ …\ PSAPTABD.

The counter ignores any relationships within RZ20's tree for the monitor set-monitor pair. Instead, it allows you to select each of these four elements using the parameter dependency feature. That is, after a monitor set is selected, the monitor list has only monitors belonging to that monitor set. After the monitor is selected, the system name parameter only has values that belong to the combination of monitor set-monitor, etc.

Performance attributes show a numeric value as the primary datapoint and any other messages as an extended datapoint.

Status and log attributes show their status value – green, yellow, red and white (normal, warning, critical and no data reported, respectively). The primary datapoint is shown as 1, 2, 3 and 0 respectively. The intelligent datapoint is available as explanation of returned status.

This counter enables monitoring of any parts of R/3 of SAP modules, which supply data to CCMS.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Monitor Set

Name of the monitor set. You can specify one or more sets. In any combination, select values from the discovered list, or enter values manually.

Monitor

Name of the monitor within the selected monitor set. You can specify one or more monitors. In any combination, select values from the discovered list, or enter values manually.

System ID

The system ID (or system name) of the monitoring system. You can specify one or more systems. In any combination, select values from the discovered list, or enter values manually.

Context

The monitored context within the system ID. You can specify one or more contexts. In any combination, select values from the discovered list, or enter values manually.

Object

The monitored object within the specified context. You can specify one or more objects. In any combination, select values from the discovered list, or enter values manually.

Name

Name of MTE from R/3's RZ20 transaction. You can specify one or more names. In any combination, select values from the discovered list, or enter values manually.

Stat Type

Select what type of data is returned. This parameter is pre-defined and single-selectable. Possible values are:

Active alerts	returns number of alerts
Alert history	returns number of alerts
Value (default)	returns MTE value

Show last minutes

Number of minutes of data history to return.

Data Point

Primary Data Point

A primary data point (PDP) is returned for each combination of parameters. The value returned is the MTE value or number of alerts, depending on the selection in the Stat type parameter. If an error is encountered during data collection, the counter returns 999.

Intelligent Data Point

The intelligent data point (IDP) provides the following information:

- MTE status
- Timestamp
- MTE Name

Interval

Not applicable.

SAP Connection Test (SM59)

This counter tests the connection to the selected remote system, as described in R/3. This is the same connection test as the R/3 transaction SM59.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Connection Name

Name of the connection described in SM59. Enter a string. There is no default value.

Data Point

Primary Data Point

The primary data point (PDP) is one of the following values:

- 0 if the test fails
- 1 if the test is successful
- 999 if the counter experiences an error during data collection

Intelligent Data Point

The intelligent data point (IDP) returns one of the following messages:

- Failure reason if the test failed
- "Connection tested OK" message if the test succeeded
- Error message if the counter encounters an error during data collection

Interval

Recommended minimum is 5 minutes.

SAP CPU Consumption

This counter monitors CPU consumption for the specific users or transactions.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

User Name

The user name to monitor. The default is all (* wildcard). You can either accept the default or enter a specific name. Wildcard characters cannot be included in the name.

TCode/Program

The user transaction code or report code to monitor. The default is all (* wildcard). You can either accept the default or enter a specific name. Wildcard characters cannot be included in the name.

Data Point

Primary Data Point

A primary data point (PDP) is returned for each combination of parameters. The value returned is the percentage of SAP CPU consumption. If an error is encountered during data collection, the counter returns 999.

If you do not specify values for the User Name or TCODE/Program parameters, the value returned is always 100%.

Intelligent Data Point

The intelligent data point (IDP) provides the following information:

Username	Name of the user.
Tcode/Program	Name of transaction code or report.
CPU (ms)	Current CPU consumption in milliseconds.
CPU (%)	Current CPU consumption in percentage.
WP-Type	Number of user sessions.

Interval

Recommended minimum is 5 minutes.

SAP Itemized Active Users

This counter returns the SAP users connected to the specified SAP instance and application servers. It is similar to the SAP active users counter, with the addition of the Application Server Name parameter (multi-selectable and wildcard enabled).

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Count

Maximum number of servers on which to report data. The default is **`10`**. The instance number can range between **`00`** and **`100`**.

Apply Operation

The monitoring level. This parameter is pre-defined and single-selectable. Possible values are:

None	A primary data point is returned for each server specified in the Application Server parameter.
Sum	The primary data point is a sum for all servers specified in the Application Server parameter.

Application Server

Name of the application server you want to monitor. You can specify one or more names. In any combination, select values from the discovered list, or enter values manually.

Data Point

Primary Data Point

The primary data point (PDP) is the current queue depth as a percentage of the defined maximum. The Apply Operation parameter determines whether the counter returns a summary data point or individual data points. If an error is encountered during data collection, the counter returns 999.

Intelligent Data Point

The intelligent data point (IDP) displays the following information:

Sysname	Full application server name.
TerminalID	Terminal identification.
Client	User's logon client number.
Username	Name of the user.
Report/Tcode	Name of tcode or report currently used by user.
Terminal	Terminal name.
Time	Dialog time.
Sessions	Number of user sessions.

Interval

Recommended minimum is 5 minutes.

SAP Itemized Job Status

This counter reports the status of jobs that meet the specified criteria. It is similar to the SAP Job Status counter, with the addition of the Apply Operation parameter and the all (* wildcard) default setting for the Job Status parameter.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Job Name

The job name to monitor. The default is all. You can either accept the default or enter a name.

User Name

The user name to monitor. The default is all. You can either accept the default or enter a name.

Job Status

The statuses you want to monitor. This parameter is predefined and multi-selectable. Possible values are:

* (all - default)

Active

Canceled

Finished

Ready

Released

Scheduled

Event Name

Name of a SAP job event. If you specify an event name for this parameter, this counter returns batch jobs related to that event only. The default is to monitor all events. You can either accept the default or enter a name.

Start Time

Number of minutes back from the current time you want this counter to monitor job entries. Specify a value from -999999 to 0 (in minutes). The default value is -60.

End Time

Number of minutes forward from the current time you want this counter to monitor job entries. Specify a value from 0 to 999999 (in minutes). The default value is 60.

Apply Operation

The monitoring level. This parameter is predefined single-selectable. Possible values are:

None	A primary data point is returned for each status type specified in the Job

	Status parameter.
Sum (default)	The primary data point is a sum for all status types specified in the Job Status parameter.

Data Point

Primary Data Point

The primary data point (PDP) is the number of jobs. The Apply Operation parameter determines whether the counter returns a summary data point or individual data points. If an error is encountered during data collection, the counter returns 999.

Intelligent Data Point

The intelligent data point (IDP) lists the following information for each job status:

! Total jobs found

! Scheduled

! Released

! Ready

! Active

! Finished

! Cancelled

The IDP also includes a table with the following information, organized by job status:

Jobname	Name of the job.
Job-count	Internal job ID.
Status	Job status.
Log	Short log messages.

Interval

Recommended minimum is 5 minutes.

SAP Itemized Spool Queue

This counter returns the current number of entries in the SAP spool queue that match the specified criteria. It is similar to the SAP Spool Queue counter, with the addition of the Apply Operation parameter and the all default setting for the Request Status parameter.

Parameter

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Request Status

Request status that you want to monitor. This parameter is predefined and multi-selectable. Possible values are:

* (all - default)

Problem

Processing

Succeeded

Without

Apply Operation

The monitoring level. This parameter is predefined and single-selectable. Possible values are:

None	A primary data point is returned for each status type specified in the Request Status parameter.
Sum (default)	The primary data point is a sum for all status types specified in the Request Status parameter.

Data Point

Primary Data Point

The primary data point (PDP) is the number of current entries in the SAP spool queue. The Apply Operation parameter determines whether the counter returns a summary data point or individual data points. If an error is encountered during data collection, the counter returns 999.

Intelligent Data Point

The intelligent data point (IDP) is the number of entries in the spool queue for each request status.

Interval

Recommended minimum is 5 minutes.

SAP Job Status

This counter reports the number of jobs that are selected that meet the criteria you specify.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Job Name

The job name to monitor. The default is all. You can either accept the default or enter a name.

User Name

The user name to monitor. The default is all. You can either accept the default or enter a name.

Job Status

The job status you want to monitor. This parameter is predefined and single-selectable. Possible values are:

* (all)

Active

Canceled

Finished

Ready

Released

Scheduled (default)

Event Name

Name of a SAP job event. If you specify an event name for this parameter, this counter returns batch jobs related to that event only. The default is to monitor all events. You can either accept the default or enter a name.

Start Time

Number of minutes back from the current time you want this counter to monitor job entries. Specify a value from -999999 to 0 (in minutes). The default value is -60.

End Time

Number of minutes forward from the current time you want this counter to monitor job entries. Specify a value from 0 to 999999 (in minutes). The default value is 60.

Data Point

Primary Data Point

A primary data point (PDP) is returned for each combination of parameters. The number returned is the number of jobs. If an error is encountered during data collection, the counter returns 999.

Intelligent Data Point

The intelligent data point (IDP) lists the following information:

- Total jobs found
- Scheduled
- Released
- Ready
- Active
- Finished
- Cancelled

The IDP also includes a table with the following information, organized by job status:

Jobname	Name of the job.
Job-count	Internal job ID.

Status	Job status.
Log	Short log messages.

Interval

Recommended minimum is 5 minutes.

SAP Memory Usage

This counter returns the total memory usage for the specified number of SAP users in the SAP system.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Count

Maximum number of users for which the counter is to report memory utilization values. Specify a value from 0 to 100. The default is **`10`**. This number is the number rows of information that is reported in the Intelligent Data Point (IDP) table (described below).

Metrics

Units in which you want memory usage returned. This parameter is predefined and single-selectable. Possible values are:

bytes

KB

MB

Data Point

Primary Data Point

The primary data point (PDP) is the total memory utilization. If an error is encountered during data collection, the counter returns 999.

Intelligent Data Point

The intelligent data point (IDP) is a table with the following information, organized by user:

Client	Client number.
User	User name or owner of the job.
TransCode	Transaction code name.
Roll Area	Size of roll area.
Page Area	Size of page area.
Shared Memory	Size of shared memory.

Heap Memory	Size of heap memory.
Summary Memory	Summary of all types of memory.
TerminalID	Terminal identification number.

Interval

Recommended minimum is 5 minutes.

SAP Number of Dumps

This counter returns the number of dumps generated by the target system in the current day (since midnight on the SAP system).

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Expression

Pattern to use to match dump's short text. The default is all

Data Point

Primary Data Point

The primary data point (PDP) is the number dumps. If an error is encountered during data collection, the counter returns 999.

Intelligent Data Point

The intelligent data point (IDP) lists the following information for each dump:

Time	Time dump was created.
Application Host	Application host name.
User	User name.
Client	Client number.
Short Text	Dump description.

Interval

Recommended minimum is 15 minutes.

SAP Page/Roll Area

This counter monitors the page or roll area statistics.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Return Value Metrics

The page and roll area metrics. This parameter is pre-defined and multi-selectable. Possible values are:

Maximum Paging Area Used (%)

Maximum Paging Area Used (KB)

Maximum Roll Area Used (%)

Maximum Roll Area Used (KB)

Size of the Paging Area (KB)

Size of the Paging Area in the Shared Memory (KB)

Size of the Paging File (KB)

Size of the Roll Area (KB)

Size of the Roll Area in the Shared Memory (KB)

Size of the Roll File (KB)

Size of the Work Process-Local Paging Buffer (KB) (default)

Used Paging Area (%)

Used Paging Area (KB)

Used Roll Area (%)

Used Roll Area (KB)

Data Point

Primary Data Point

A primary data point (PDP) is returned for each combination of parameters. The value returned is the value of the specified metric (KB or %). The counter returns 999 if it encounters an error during data collection.

Intelligent Data Point

The intelligent data point (IDP) lists the statistics for all page and roll metrics.

Interval

Recommended minimum is 5 minutes.

SAP Page/Roll Area Max

This counter returns the maximum page or roll area statistics for the specified task interval.

Determining a Statistic's Maximum Value

A Remote Function Call (RFC) is made at each task interval to get the data. It searches the internal cache for the previously stored value of the same metric with a timestamp within the time range specified with the “Period in min” parameter.

If a value is found for the specified metric, it is compared with the current value. The SAP Page/Roll Area Max counter returns the greater of the two values and stores it in the cache with current timestamp.

If a stored value is not found for the specified metric, the cache is cleared and the current value is stored in it. The counter returns this value.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Return Value Metrics

The page and roll area metrics. This parameter is predefined multi-selectable. Possible values are:

Maximum Paging Area Used (%)

Maximum Paging Area Used (KB)

Maximum Roll Area Used (%)

Maximum Roll Area Used (KB)

Size of the Paging Area (KB)

Size of the Paging Area in the Shared Memory (KB)

Size of the Paging File (KB)

Size of the Roll Area (KB)

Size of the Roll Area in the Shared Memory (KB)

Size of the Roll File (KB)

Size of the Work Process-Local Paging Buffer (KB) (default)

Used Paging Area (%)

Used Paging Area (KB)

Used Roll Area (%)

Used Roll Area (KB)

Period in min

Specify a maximum duration of time in minutes. The default is 60.

Data Point

Primary Data Point

A primary data point (PDP) is returned for each combination of parameters. The value returned is the maximum value of the specified metric (KB or %), determined by the method described in Determining a statistic's maximum value. The counter returns 999 if it encounters an error during data collection.

Intelligent Data Point

The intelligent data point (IDP) lists, for the period, the maximum values for all statistics.

Interval

Recommended minimum is 5 minutes.

SAP Process Monitoring

This counter returns CPU utilization or memory usage for selected processes. These processes must be set up to be monitored by the SAP Operation System Collector.

To gather this data from the target R/3 instances, you must set up SAP OS Collector (saposcol) to gather information about system processes. Complete instructions are described in the document called "Operation System Collector SAPOSCOL: Propertied, Operation and Installation". It is available from the SAP web site.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable from the discovered list.

Process Pattern or User Pattern

Process or user to monitor. This parameter is single-selectable from the dynamically discovered list.

Metrics

Usage metrics for monitoring. This parameter is predefined and multi-selectable. Possible values are:

CPU Utilization (%)

Process Count (default)

Resident Size (KB)

VM Size (KB)

Data Point

Primary Data Point

A primary data point (PDP) is returned for each combination of parameters. The returned value is the value of the selected metric. If an error is encountered during data collection, the counter returns 999.

Intelligent Data Point

The intelligent data point (IDP) displays the following information for each metric:

Last SapOsCol sample was taken at: <date><time>
SapOsCol collection interval: <number_of_seconds> sec.

Interval

Recommended minimum is 5 minutes.

SAP Spool Queue

This counter returns the current number of SAP spool queue entries that match the specified criteria.

Parameter

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Request Status

Request status that you want to monitor. This parameter is predefined and single-selectable. Possible values are:

Problem

Processing (default)

Succeeded

Without

Data Point

Primary Data Point

The primary data point (PDP) for this counter is the number of current SAP spool queue entries that match the specified criteria. If an error is encountered during data collection, the counter returns 999.

Intelligent Data Point

The intelligent data point (IDP) is the number of entries in the spool queue for each request status.

Interval

Recommended minimum is 5 minutes.

SAP System Log Entries

This counter returns, for the selected time period, the entries that match the specified expression.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Time Period (Minutes)

Number of minutes back from the current time you want this counter to monitor job entries. Specify a value from 5 to 180 (in minutes). The default value is 60.

Expression

Pattern to use to match the message text in the SAP system log . The default is all. You can either accept the default or enter a string.

Data Point

Primary Data Point

The primary data point (PDP) is the number of entries in the SAP system log that match the selection criteria.

Intelligent Data Point

The intelligent data point (IDP) displays the following information for each message:

Severity	Message severity level: Error, Warning, or Normal.
Time	Message time.
Type	Work process type and number.
PID	System process identifier of the work process.
Client	Client number.
User	User name.
Tcode	Transaction code.
Mno	Message number.
Text	Message text.

Interval

Recommended minimum is 10 minutes.

SAP Top CPU Utilization

This counter returns the highest CPU utilization, by process, for the top 40 processes on the SAP R/3 application server.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Data Point

Primary Data Point

The primary data point (PDP) is the highest CPU utilization value. If an error is encountered during data collection, the counter returns 999.

Intelligent Data Point

The intelligent data point (IDP) displays the following information for the top 40 processes:

PID	System process identifier.
Instance	Name of SAP R/3 instance.
Command	System process name.
CPU Util[%]	CPU utilization value.
CPU Time[s]	CPU time value.
Working Set[KB]	Top physical memory that is assigned to the process.
Private Pages[KB]	Total of the entire memory (physical and virtual) that is assigned to the process (Windows systems only, this value is 0 on UNIX).
Prior	Process priority.

Interval

Recommended minimum is 5 minutes.

SAP Top Load

This counter returns a maximum workload statistic for the SAP system.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Count

Maximum number of active servers monitored for workload information. Valid entries are from 0 to 100. The default is **10**.

Sorting Parameter

Workload characteristic you want to monitor. This parameter is predefined and multi-selectable. Possible values are:

CPU Time

DB Time

Response Time (default)

Transfer Size

Wait Time

Time Metrics

Units for monitoring CPU Time, DB Time, and Response Time, and Wait Time. This parameter is predefined and single-selectable. Possible values are:

MilliSeconds (default)

Seconds

Size Metrics

Unit of space for monitoring Transfer Size. This parameter is predefined and single selectable. Possible values are:

Bytes

KiloBytes (default)

MegaBytes

Data Point

Primary Data Point

A primary data point (PDP) is returned for each combination of parameters. The returned value is the top workload, in time or size. If an error is encountered during data collection, the counter returns 999.

Intelligent Data Point

The intelligent data point (IDP) displays the following information for each transaction:

User	User name.
Transaction	Transaction code name.
Report	Report name.
Background Job Name	Name of background job, if valid.
Task Type	Type of the task.
Response Time	Response name.
CPU Time	CPU time.
Wait Time	Wait time.
DB Time	Database time.
Transfer Size	Number of transferred bytes.

Interval

Recommended minimum is 10 minutes.

SAP User Function Call

This counter calls any R/3 RFC-enabled function when it is designed according the following ServerVantage rules. This counter enables you to create and implement your own custom SAP R/3 counters.

ServerVantage User Function Call Guidelines

- The function name can be any character string.
- The function should have one import, one export, and one table parameter.

  Import parameter: SV_PARAMETERS is a character string that serves for passing data from ServerVantage to R/3 function. You define how this string is parsed in R/3 function.

  Export parameter: SV_VALUE must be float type and serves for passing data point values from R/3 function to the Java Agent.

  Table parameter: SV_EXTENDED_DP is an optional parameter that serves for passing intelligent (extended) data points from R/3 function to Java Agent. It can be any character string. To pass intelligent data points, you need to include the table header.

- Parameter names cannot be changed.

  Exceptions: You can define any number of exceptions. In the case of an exception within RFM, the Monitoring tree displays –1 in the primary data Point (PDP) and an exception message in Intelligent Data Point (IDP).

- In the body of the function, you may use any manipulations to retrieve data from R/3 and set SV_VALUE and SV_EXTENDED_DP.
- The function MUST NOT have any GUI or screen output statements, or any statements requiring dialog, interaction, or additional answers.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Function Name

R/3 Remote Function Module (RFM) name.

Parameters

Parameters to pass to the function.

Data Point

Primary Data Point

The primary data point (PDP) is the value returned by the R/3 RFM. If an error is encountered during data collection, the counter returns 999.

Intelligent Data Point

The intelligent data point (IDP) is returned by the R/3 RFM.

Interval

Not applicable.

SAP Work Processes

This counter returns the number of work processes running on a SAP instance according to the specified criteria.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Process Type

Type of work process. This parameter is predefined and multi-selectable. Possible values are:

BDG	Background
DIA (default)	Dialog
ENQ	Enqueue
SPO	Spool
UP2	Update 2
UPD	Update

Process State

Process state to monitor. This counter is predefined and multi-selectable. Possible values are:

Completed

Running

Stopped

Waiting (default)

Data Point

Primary Data Point

A primary data point (PDP) is returned for each combination of parameters. The value returned is the number of work processes. If an error is encountered during data collection, the counter returns 999.

Intelligent Data Point

The intelligent data point (IDP) lists the following process information:

- Total Work Processes
- Work Processes - Waiting State
- Work Processes - Running State
- Work Processes - Stopped State
- Work Processes - Completed State

The IDP also includes a table with the following information, for each process:

Number	Process sequential number.
Type	Process type.
Process ID	Process system ID.

Status	Process status.
Reason	Work process is waiting.
Semaphore	Semaphore for which the work process is waiting.
Restart	Restart work process after dump.
Dumps	Number of dumps.
CPU	CPU time.
Elapsed Time	Previous execution time of request (elapsed).
Client	Client number.
User	User that is using the process.
Report	Report or tcode name used by the process.
Action	What the process is doing.
Table	Database table last accessed by the work process.

Interval

Recommended minimum is 10 minutes.

SAP Workload Statistic

This counter returns selected ST03 workload statistics for selected task types.

Parameters

SAP Instance

Composite name of the SAP R/3 instance you want to monitor. This parameter is single-selectable. Select an instance from the discovered list.

Task Type

The task type you want to monitor. This parameter is predefined and multi-selectable. Possible values are:

ALE

AUTOABA (default)

BCKGRD

BUF.SYN

DIALOG

ENQUEUE

FTP

HTTP

HTTPS

NNTP

RFC

SMTP

SPOOL

UPDATE

UPDATE2

Statistic Name

The workload statistic. This parameter is predefined and multi-selectable. Possible values are:

CPU time avg (ms)

CPU time total(s)

Database calls

Database requests

Database requests: Changes

Database requests: Direct reads

Database requests: Sequential reads

DB time avg (ms)

DB time total(s)

Dialog Steps (default)

Dialog steps/s

Frontend net time avg (ms)

Frontend net time total(s)

GUI time avg (ms)

GUI time total(s)

Requested kBytes

Response time avg(ms)

Response time total(s)

Roll in time

Roll ins

Roll out time

Roll outs

Roll wait time

Time per DB request

Time per DB request: Changes and commits

Time per DB request: Direct reads

Time per DB request: Sequential reads

Wait time avg (ms)

Wait time total(s)

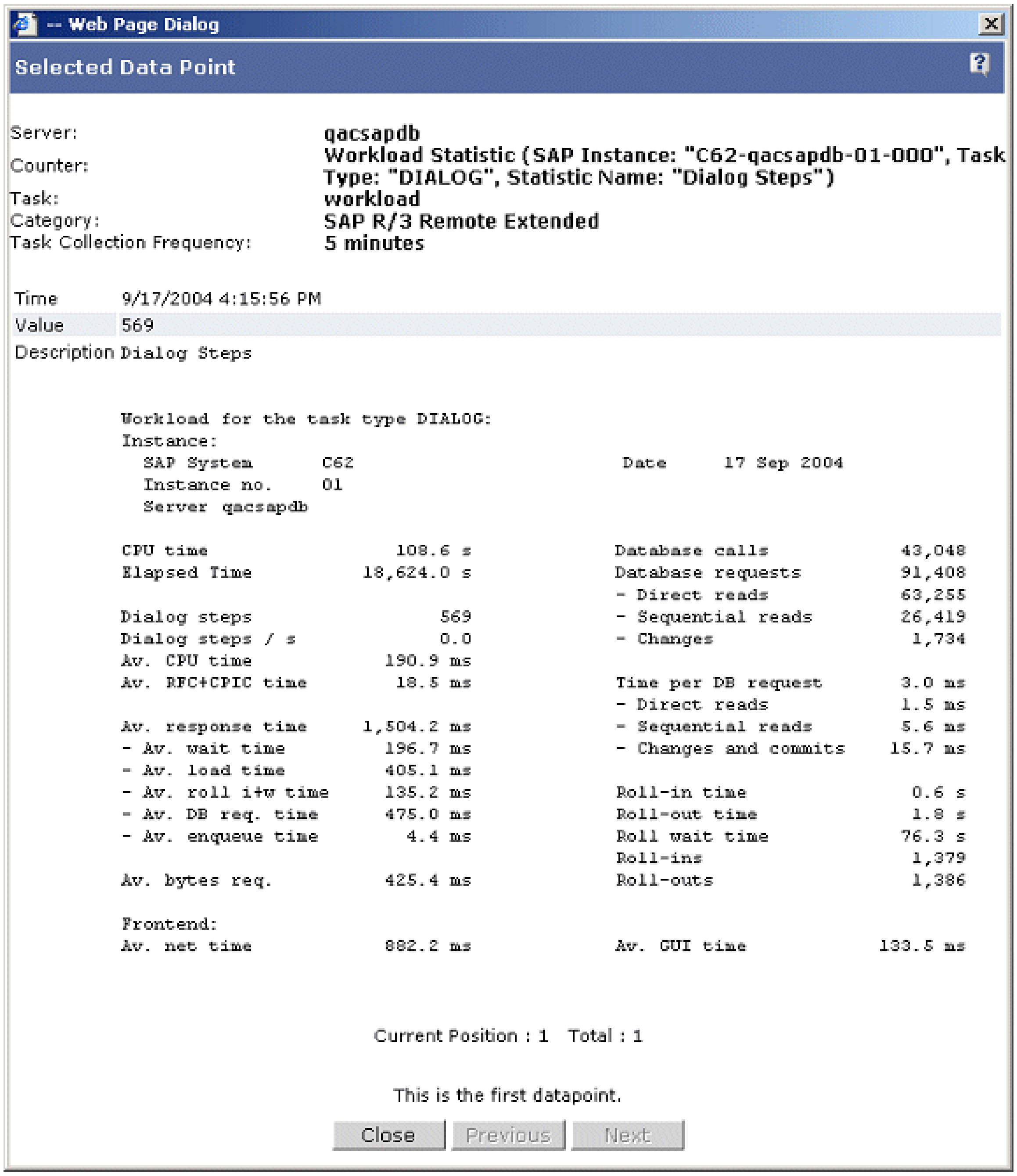
Data Point

Primary Data Point

A primary data point (PDP) is returned for each combination of parameters. The value returned is the selected workload statistic. If an error is encountered during data collection, the counter returns 999.

Intelligent Data Point

The intelligent data point (IDP) displays a list of the remaining statistics, for example:



Server: qacsapdb
Counter: Workload Statistic (SAP Instance: "C62-qacsapdb-01-000", Task Type: "DIALOG", Statistic Name: "Dialog Steps")
Task: workload
Category: SAP R/3 Remote Extended
Task Collection Frequency: 5 minutes

Time	9/17/2004 4:15:56 PM
Value	569
Description	Dialog Steps

```
Workload for the task type DIALOG:
Instance:
  SAP System      C62                        Date     17 Sep 2004
  Instance no.    01
  Server  qacsapdb

CPU time               108.6 s          Database calls            43,048
Elapsed Time        18,624.0 s          Database requests         91,408
                                        - Direct reads            63,255
Dialog steps               569          - Sequential reads        26,419
Dialog steps / s           0.0          - Changes                  1,734
Av. CPU time          190.9 ms
Av. RFC+CPIC time      18.5 ms          Time per DB request       3.0 ms
                                        - Direct reads            1.5 ms
Av. response time   1,504.2 ms          - Sequential reads        5.6 ms
- Av. wait time       196.7 ms          - Changes and commits    15.7 ms
- Av. load time       405.1 ms
- Av. roll i+w time   135.2 ms          Roll-in time               0.6 s
- Av. DB req. time    475.0 ms          Roll-out time              1.8 s
- Av. enqueue time      4.4 ms          Roll wait time            76.3 s
                                        Roll-ins                   1,379
Av. bytes req.        425.4 ms          Roll-outs                  1,386

Frontend:
Av. net time          882.2 ms          Av. GUI time            133.5 ms
```

Current Position : 1   Total : 1

This is the first datapoint.

Interval

Recommended minimum is 60 minutes.

SNMP Counters

SNMP Counters

SNMP Remote Monitoring uses the SNMP service to provide network and system counters. SNMP counters can be retrieved from any machine that is running an SNMP server. QALoad uses the default SNMP port (161) and default Community (public). If necessary, you can enter a different port and community in the Configure Monitor Dialog screen when you create or edit an SNMP monitoring task and when you create or edit an SNMP monitoring template. Although SNMP does not require a user name and password, the SNMP agent must be configured to allow read-only access from the Conductor machine.

In addition to the standard SNMP counters supported by QALoad Remote Monitoring, you can create your own custom Object Identifier (OID) file with counters you define. See Customizing SNMP Counter Discovery for more information.

Standard SNMP Counters

Standard SNMP counters that are supported by QALoad Remote Monitoring are categorized below.

ICMP

icmpInMsgs/sec: the rate at which ICMP messages are received
icmpInErrors: the number of ICMP messages received having ICMP errors
IcmpInDestUnreachs: the number of ICMP Destination Unreachable messages received
IcmpInTimeExcds: the number of ICMP Time Exceeded messages received
IcmpInParmProbs: the number of ICMP Parameter Problem messages received

IcmpInSrcQuenchs: the number of ICMP Source Quench messages received
icmpInRedirects/sec: the rate at which ICMP Redirect messages are received
icmpInEchos/sec: the rate at which ICMP Echo messages are received
icmpInEchoReps/sec: the rate at which ICMP Echo Reply messages are received
icmpInTimestamps/sec: the rate at which ICMP Timestamp messages are received
icmpInTimestampReps/sec: the rate at which ICMP Timestamp Reply messages are received
icmpInAddrMasks: the number of ICMP Address Mask Request messages received
icmpInAddrMaskReps: the number of ICMP Address Mask Reply messages received
icmpOutMsgs/sec: the rate at which ICMP messages are sent
icmpOutMsgs/sec: the number of ICMP messages not sent due to ICMP errors
icmpOutDestUnreachs: the number of ICMP Destination Unreachable messages sent
icmpOutTimeExcds: the number of ICMP Time Exceeded messages sent
icmpOutParmProbs: the number of ICMP Parameter Problem messages sent
icmpOutSrcQuenchs: the number of ICMP Source Quench messages sent
icmpOutRedirects/sec: the number of ICMP Redirect messages sent
icmpOutEchos/sec: the number of ICMP Echo messages sent
icmpOutEchoReps/sec: the number of ICMP Echo Reply messages sent
icmpOutTimestamps/sec: the number of ICMP Timestamp messages sent
icmpOutTimestampReps/sec: the number of ICMP Timestamp Reply messages sent
icmpOutAddrMasks: the number of ICMP Address Mask Request messages sent
icmpOutAddrMaskReps: the number of ICMP Address Mask Reply messages sent

IP

ipForwarding: the indication of whether this entity is acting as an IP router in respect to the forwarding of datagrams received by, but not addressed to, this entity.
ipDefaultTTL: the default value inserted into the Time-To-Live field of the IP header of datagrams originated at this entity, whenever a TTL value is not supplied by the transport layer protocol.
ipInReceives/sec: the rate of input datagrams received from interfaces, including those received in error.
ipInHdrErrors: the number of input datagrams discarded due to errors in their IP headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options, and so on.
ipInAddrErrors: the number of input datagrams discarded because the IP address in their IP header's destination field was not a valid address to be received at this entity.
ipForwDatagrams/sec: the rate of input datagrams for which this entity was not their final IP destination, as a result of which an attempt was made to find a route to forward them to that final destination.
ipInUnknownProtos: the number of locally-addressed datagrams receive successfully but discarded because of an unknown or unsupported protocol.

ipInDiscards: the number of input IP datagrams for which no problems were encountered to prevent their continued processing, but which were discarded (for example, for lack of buffer space).
ipInDelivers/sec: the rate of input datagrams successfully delivered to IP user-protocols (including ICMP).
ipOutRequests: the number of IP datagrams which local IP user-protocols (including ICMP) supplied to IP in requests for transmission.
ipOutDiscards: the number of output IP datagrams for which no problem was encountered to prevent their transmission to their destination, but which were discarded (for example, for lack of buffer space).
ipOutNoRoutes: the number of IP datagrams discarded because no route could be found to transmit them to their destination.
ipReasmTimeout: the maximum number of seconds which received fragments are held while they are awaiting reassembling at this entity.
ipReasmReqds: the number of IP fragments received which needed to be reassembled at this entity.
ipReasmOKs: the number of IP datagrams successfully re-assembled.
ipReasmFails: the number of failures detected by the IP re-assembly algorithm (for whatever reason: timed out, errors, etc).
ipFragOKs: the number of IP datagrams that have been successfully fragmented at this entity.
ipFragFails: the number of IP datagrams that have been discarded because they needed to be fragmented at this entity but could not be, for example, because their Don't Fragment flag was set.
ipFragCreates/sec: the rate of IP datagram fragments that have been generated as a result of fragmentation at this entity.
ipRoutingDiscards: the number of routing entries which were chosen to be discarded even though they are valid.

SNMP

snmpInPkts/sec: the rate of messages delivered to the SNMP entity from the transport service.
snmpOutPkts/sec: the rate at which SNMP Messages were passed from the SNMP protocol entity to the transport service.
snmpInBadVersions: the number of SNMP messages which were delivered to the SNMP entity and were for an unsupported SNMP version.
snmpInBadCommunityNames: the number of SNMP messages delivered to the SNMP entity which used a SNMP community name not known to said entity.
snmpInBadCommunityUses: the number of SNMP messages delivered to the SNMP entity which represented an SNMP operation which was not allowed by the SNMP community named in the message.
snmpInASNParseErrs: the number of ASN.1 or BER errors encountered by the SNMP entity when decoding received SNMP messages.
snmpInTooBigs: the number of SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is tooBig.
snmpInNoSuchNames: the number of SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is noSuchName.
snmpInBadValues: the number of SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is badValue.
snmpInReadOnlys: the number valid SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is readOnly.
snmpInGenErrs: the number of SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is genErr.
snmpInTotalReqVars/sec: the rate of MIB objects which have been retrieved successfully by the SNMP protocol entity as the result of receiving valid SNMP Get-Request and Get-Next PDUs.
snmpInTotalSetVars/sec: the rate of MIB objects which have been altered successfully by the SNMP protocol entity as the result of receiving valid SNMP Set-Request PDUs.
snmpInGetRequests/sec: the rate of SNMP Get-Request PDUs which have been accepted and processed by the SNMP protocol entity.
snmpInGetNexts/sec: the rate of SNMP Get-Next PDUs which have been accepted and processed by the SNMP protocol entity.
snmpInSetRequests/sec: the rate of SNMP Get-Response PDUs which have been accepted and processed by the SNMP protocol entity.

snmpInGetResponses/sec: the rate of SNMP Set-Request PDUs which have been accepted and processed by the SNMP protocol entity.
snmpInTraps: the number of SNMP Trap PDUs which have been accepted and processed by the SNMP protocol entity.
snmpOutTooBigs: the number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status field is tooBig.
snmpOutNoSuchNames: the number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status is noSuchName.
snmpOutBadValues: the number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status field is badValue.
snmpOutGenErrs: the number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status field is genErr.
snmpOutGetRequests/sec: the rate of SNMP Get-Request PDUs which have been generated by the SNMP protocol entity.
snmpOutGetNexts/sec: the rate of SNMP Get-Next PDUs which have been generated by the SNMP protocol entity.
snmpOutSetRequests/sec: the rate of SNMP Set-Request PDUs which have been generated by the SNMP protocol entity.
snmpOutGetResponses/sec: the rate of SNMP Get-Response PDUs which have been generated by the SNMP protocol entity.
snmpOutTraps: the number of SNMP Trap PDUs which have been generated by the SNMP protocol entity.
snmpOutTraps: indicates whether the SNMP entity is permitted to generate authenticationFailure traps.

TCP

tcpRtoAlgorithm: the algorithm used to determine the timeout value used for retransmitting unacknowledged octets.
tcpRtoMin: the minimum value permitted by a TCP implementation for the retransmission timeout.
tcpRtoMax: the maximum value permitted by a TCP implementation for the retransmission timeout.
tcpMaxConn: the limit on the total number of TCP connections the entity can support.
tcpActiveOpens: the number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state.
tcpAttemptFails: the number of times TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state.
tcpEstabResets: the number of times TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state.
tcpCurrEstab: the number of TCP connections for which the current state is either ESTABLISHED or CLOSE-WAIT.
tcpInSegs/sec: the rate at which segments are received, including those received in error.
tcpOutSegs/sec: the rate at which segments are sent, including those on current connections but excluding those containing only retransmitted octets.
tcpRetransSegs/sec: the rate at which segments are retransmitted.
tcpInErrs/sec: the rate at which segments are received in error.
tcpOutRsts/sec: the rate at which segments containing the RST flag are sent.
tcpPassiveOpens: the total number of times TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state.

UDP

udpInDatagrams/sec: the rate of UDP datagrams being delivered to UDP users.
udpNoPorts/sec: the rate of received UDP datagrams for which there was no application at the destination port.
udpInErrors: the number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.
udpOutDatagrams/sec: the rate at which UDP datagrams are sent.

Solaris: Sun System

Collisions/sec: the rate of output collisions.
CpuUser%: the percentage of non-idle processor time that is spent in user mode.
CpuNice%: the percentage of non-idle processor time that is spent in nice mode.
CpuSys%: the percentage of non-idle processor time that is spent in system mode.
CpuIdle%: the percentage of idle processor time.
IfInPackets/sec: the rate of input packets.
IfOutPackets/sec: the rate of output packets.
IfInErrors: the total number of input errors.
IfOutErrors: the total number of output errors.
Interrupts/sec: the rate of system interrupts.
PagesIn KBytes/sec: the rate of pages read in from disk.
PagesOut KBytes/sec: the rate of pages written to disk.
SwapIn KBytes/sec: the rate at which pages are being swapped in.
SwapOut KBytes/sec: the rate at which pages are being swapped out.

HP-UX: HP System

AvgJobs1: the average number of jobs in the last minute * 100.
AvgJobs5: the average number of jobs in the last 5 minutes * 100.
AvgJobs15: the average number of jobs in the last 15 minutes * 100.
CpuUser%: the percentage of non-idle processor time that is spent in user mode.
CpuNice%: the percentage of non-idle processor time that is spent in nice mode.
CpuSys%: the percentage of non-idle processor time that is spent in system mode.
CpuIdle%: the percentage of idle processor time.
FreeMemory KBytes: the amount of idle memory.
FreeSwap KBytes: the amount of free swap space on the system.
MaxProc: the maximum number of processes allowed.
MaxUserMem KBytes: the amount of maximum user memory on the system.
PhysMemory KBytes: the amount of physical memory on the system.
Users: the number of users logged on to the machine.

LINUX Memory

AvailableSwap KBytes: the available swap on the system.
Buffered KBytes: the amount of memory used as buffers.
Cached KBytes: the amount of memory cached.
FreeMemory KBytes: the amount of idle memory.
Shared KBytes: the amount of memory shared.
TotalMemory KBytes: the total amount of memory on the system.
TotalSwap KBytes: the total swap size for the system.

LINUX System

CpuUser%: the percentage of non-idle processor time that is spent in user mode.
CpuNice%: the percentage of non-idle processor time that is spent in nice mode.
CpuSys%: the percentage of non-idle processor time that is spent in system mode.
CpuIdle%: the percentage of idle processor time.

Windows HTTP Server

httpTotalFilesSent: the total number of files sent by this HTTP server.
httpTotalFilesReceived: the total number of files received by this HTTP server.
httpCurrentAnonymousUsers: the number of anonymous users currently connected to this HTTP server.
httpCurrentNonAnonymousUsers: the number of non-anonymous users currently connected to this HTTP server.
httpTotalAnonymousUsers: the total number of anonymous users that have ever connected to this HTTP

server.
httpTotalNonAnonymousUsers: the total number of non-anonymous users that have ever connected to this HTTP server.
httpMaximumAnonymousUsers: the maximum number of anonymous users simultaneously connected to this HTTP server.
httpMaximumNonAnonymousUsers: the maximum number of non-anonymous users simultaneously connected to this HTTP server.
httpCurrentConnections: the current number of connections to the HTTP server.
httpMaximumConnections: the maximum number of simultaneous connections to the HTTP server.
httpConnectionAttempts: the total number of connection attempts to the HTTP server.
httpLogonAttempts: the total number of logon attempts to the HTTP server.
httpTotalOptions: the total number of requests made to this HTTP server using the OPTIONS method.
httpTotalGets: the total number of requests made to this HTTP server using the GET method.
httpTotalPosts: the total number of requests made to this HTTP server using the POST method.
httpTotalHeads: the total number of requests made to this HTTP server using the HEAD method.
httpTotalPuts: the total number of requests made to this HTTP server using the PUT method.
httpTotalDeletes: the total number of requests made to this HTTP server using the DELETE method.
httpTotalTraces: the total number of requests made to this HTTP server using the TRACE method.
httpTotalMove: the total number of requests made to this HTTP server using the MOVE method.
httpTotalCopy: the total number of requests made to this HTTP server using the COPY method.
httpTotalMkcol: the total number of requests made to this HTTP server using the MKCOL method.
httpTotalPropfind: the total number of requests made to this HTTP server using the PROPFIND method.
httpTotalProppatch: the total number of requests made to this HTTP server using the PROPPATCH method.
httpTotalSearch: the total number of requests made to this HTTP server using the MS-SEARCH method.
httpTotalLock: the total number of requests made to this HTTP server using the LOCK method.
httpTotalUnlock: the total number of requests made to this HTTP server using the UNLOCK method.
httpTotalOthers: the total number of requests made to this HTTP server not using the OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, MOVE, MKCOL, PROPFIND, PROPPATCH, MS-SEARCH, LOCK or UNLOCK methods.
httpCurrentCGIRequests: the number of Common Gateway Interface requests currently being serviced by this HTTP server.
httpCurrentBGIRequests: the number of Binary Gateway Interface requests currently being serviced by this HTTP server.
httpTotalCGIRequests: the total number of Common Gateway Interface requests made to this HTTP server.
httpTotalBGIRequests: the total number Binary Gateway Interface requests made to this HTTP server.
httpMaximumCGIRequests: the maximum number of Common Gateway Interface requests simultaneously processed by this HTTP server.
httpMaximumBGIRequests: the maximum number of Binary Gateway Interface requests simultaneously processed by this HTTP server.
httpCurrentBlockedRequests: the current number of requests being temporarily blocked by this HTTP server.
httpTotalBlockedRequests: the total number of requests that have been temporarily blocked by this HTTP server.
httpTotalRejectedRequests: the total number of requests that have been rejected by this HTTP server.

Windows FTP Server

ftpTotalFilesSent: the total number of files sent by this FTP server.
ftpTotalFilesReceived: the total number of files received by this FTP server.
ftpCurrentAnonymousUsers: the number of anonymous users currently connected to this FTP server.
ftpCurrentNonAnonymousUsers: the number of non-anonymous users currently connected to this FTP server.
ftpTotalAnonymousUsers: the total number of anonymous users that have ever connected to this FTP server.

ftpTotalNonAnonymousUsers: the total number of non-anonymous users that have ever connected to this FTP server.
ftpMaximumAnonymousUsers: the maximum number of anonymous users simultaneously connected to this FTP server.
ftpMaximumNonAnonymousUsers: the maximum number of non-anonymous users simultaneously connected to this FTP server.
ftpCurrentConnections: the current number of connections to the FTP server.
ftpMaximumConnections: the maximum number of simultaneous connections to the FTP server.
ftpConnectionAttempts: the total number of connection attempts to the FTP server.
ftpLogonAttempts: the total number of logon attempts to the FTP server.

Customizing SNMP Counter Discovery

QALoad currently has a standard list of Object Identifiers (OID) that it searches when discovering SNMP counters. You can create and import a list of additional OIDs in an XML file. The XML file, called the Custom OID File, introduces additional SNMP counters that you want to include during the counter discovery phase when you create or edit tasks and templates.

Note: You can include multiple categories and multiple counters in the custom OID file.

Once you create the custom OID file, you select it using the Custom OID File field in the Configure Monitor Dialog screen of the monitoring wizards. Since there is a one to one relationship between monitoring tasks and the custom file, multiple files can exist. These reside in a common, default location: `QALoad\Monitoring\SNMP`. The browse button for the Custom OID File field defaults to this location. This directory also contains two files to assist you in developing your Custom OID File:

- OID structure.xml - a template you can use for creating your custom supplemental file. This contains the basic XML structure you need to create your own Custom OID file. The Table of Custom OID File Elements describes and gives the rules for each of the XML elements in the structure.
- OID example.xml - a custom OID XML file structure. This is a sample Custom OID File.

Custom OID Template File

Use the following template, located in QALoad\Monitoring\SNMP\OID structure.xml, to create your custom OID file:

Table of Custom OID File Elements

The following table shows each element in the XML file structure, and describes its purpose and the rules that apply.

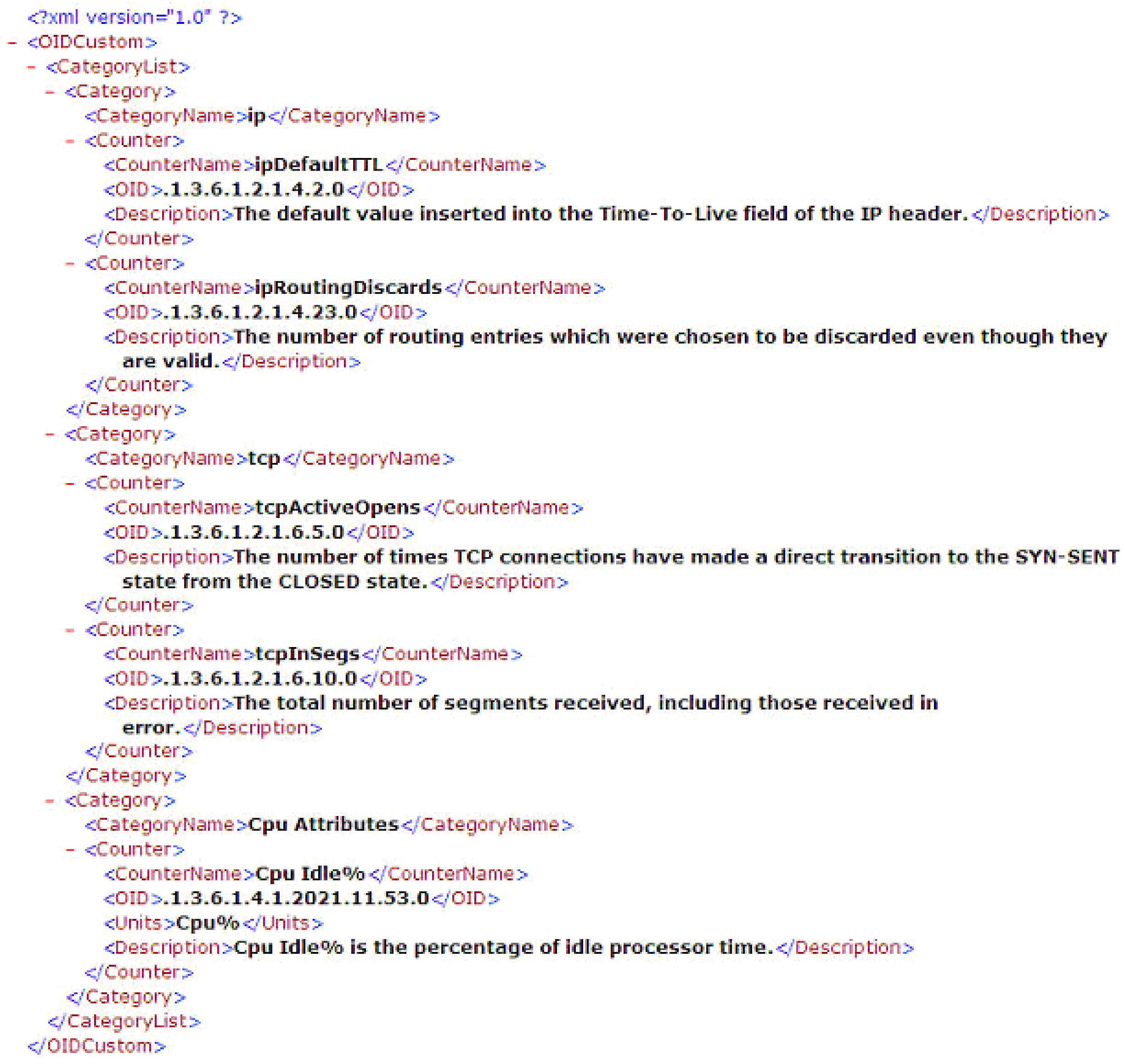
Tag	Description	Rules
<?xml version="1.0" ?>	XML file header indicating files structural version.	Must be first element of file. Only one per file.
<OIDCustom>	Begin bracket for entire OID custom information. Used to both identify content type and begin/end of file.	Only one per file. Must immediately follow XML version header.
<CategoryList>	Begin bracket for list of categories defined in this file. Used to help group all the contained categories.	Only one per file. Must immediately follow <OIDCustom> tag.

<Category>	Begin bracket for an individual category within the category list. Used to help group individual categories.	One required per category group. First instance must immediately follow <CategoryList> tag. Subsequent instances immediately follow </Category> end tag of previous group.
<CategoryName>?</CategoryName>	The display name for the category group. The "?" represents where the user defines the category name.	One required per category. Must immediately follow <Category> tag.
<Counter>	Begin bracket for individual counter information contained in a category. Multiple counters can exist per category.	One required per counter. First instance must immediately follow </CategoryName> tag. Subsequent instances immediately follow </Counter> tag.
<CounterName>?</CounterName>	The display name for the counter. The "?" represents where the user defines the counter name.	One required per counter. Only one allowed per counter.
<OID>?</OID>	The OID for the counter. The OID must start with a period "." (see example below). The "?" represents where the user defines the OID.	One required per counter. Only one allowed per counter.
<Units>?</Units>	The optional calculation units for the counter. The "?" represents where the user defines the calculation type which is one of the following: "/sec" – for counters that must take into account the time between samplings. "Cpu%" – for counters that must take into account the number of processors. A counter for which this applies is CpuIlde%. "Kbits/sec" – for converting counters that natively report in Kbit/sec to Kbytes/sec.	Optionally, one per counter. Only one allowed per counter.

<Description>?</Description>	The optional description for the counter. The "?" represents where the user defines the OID.	Optionally, one per counter. Only one allowed per counter.
</Counter>	The end bracket for an individual counter information set. Used to denote the end of individual counters attribute information.	One required per counter.
</Category>	The end bracket for an individual category, used to denote the end of a category and all of its associated counters.	One required per category.
</CategoryList>	End bracket for list of categories.	Only one per file. Must immediately precede <OIDCustom> tag.
</OIDCustom>	End bracket for entire OID custom information. Used to identify the end of file.	Only one per file. Must be last file element.

Custom OID Sample File

The sample Custom OID File shown here is located in QALoad\Monitoring\SNMP\OID example.xml:



```
  <?xml version="1.0" ?>
- <OIDCustom>
  - <CategoryList>
    - <Category>
        <CategoryName>ip</CategoryName>
      - <Counter>
          <CounterName>ipDefaultTTL</CounterName>
          <OID>.1.3.6.1.2.1.4.2.0</OID>
          <Description>The default value inserted into the Time-To-Live field of the IP header.</Description>
        </Counter>
      - <Counter>
          <CounterName>ipRoutingDiscards</CounterName>
          <OID>.1.3.6.1.2.1.4.23.0</OID>
          <Description>The number of routing entries which were chosen to be discarded even though they
            are valid.</Description>
        </Counter>
      </Category>
    - <Category>
        <CategoryName>tcp</CategoryName>
      - <Counter>
          <CounterName>tcpActiveOpens</CounterName>
          <OID>.1.3.6.1.2.1.6.5.0</OID>
          <Description>The number of times TCP connections have made a direct transition to the SYN-SENT
            state from the CLOSED state.</Description>
        </Counter>
      - <Counter>
          <CounterName>tcpInSegs</CounterName>
          <OID>.1.3.6.1.2.1.6.10.0</OID>
          <Description>The total number of segments received, including those received in
            error.</Description>
        </Counter>
      </Category>
    - <Category>
        <CategoryName>Cpu Attributes</CategoryName>
      - <Counter>
          <CounterName>Cpu Idle%</CounterName>
          <OID>.1.3.6.1.4.1.2021.11.53.0</OID>
          <Units>Cpu%</Units>
          <Description>Cpu Idle% is the percentage of idle processor time.</Description>
        </Counter>
      </Category>
    </CategoryList>
  </OIDCustom>
```

WebLogic7/8 Counters

WebLogic Remote Extended Counters

The following dynamically discovered WebLogic remote extended counter categories are provided in QALoad. Each category provides counters that extend the monitoring of your WebLogic system. The categories, counter names, and parameters are all dynamically discovered by processing the set of MBeans available in the WebLogic JMX Server.

WebLogic Application Runtime	WebLogic JMS Session Runtime
WebLogic Connector Service Runtime	WebLogic JTA Recovery Runtime
WebLogic Deployer Runtime	WebLogic JTA Runtime
WebLogic Domain Log Handler Runtime	WebLogic JVM Runtime
WebLogic Domain Runtime	WebLogic Log Broadcaster Runtime
WebLogic EJB Cache Runtime	WebLogic Message Driven EJB Runtime

WebLogic EJB Component Runtime

WebLogic EJB Locking Runtime

WebLogic EJB Pool Runtime

WebLogic EJB Transaction Runtime

WebLogic Entity EJB Runtime

WebLogic Execute Queue Runtime

WebLogic JDBC Connection Pool Runtime

WebLogic JMS Connection Runtime

WebLogic JMS Consumer Runtime

WebLogic JMS Destination Runtime

WebLogic JMS Runtime

WebLogic JMS Server Runtime

WebLogic Migratable Service Coordinator Runtime

WebLogic Server Life Cycle Runtime

WebLogic Server Runtime

WebLogic Server Security Runtime

WebLogic Servlet Runtime

WebLogic Stateful EJB Runtime

WebLogic Stateless EJB Runtime

WebLogic Time Service Runtime

WebLogic Transaction Resource Runtime

WebLogic Web App Component Runtime

WebLogic Web Server Runtime

WebLogic Application Runtime

The WebLogic Application Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered	Boolean	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Application

The application name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

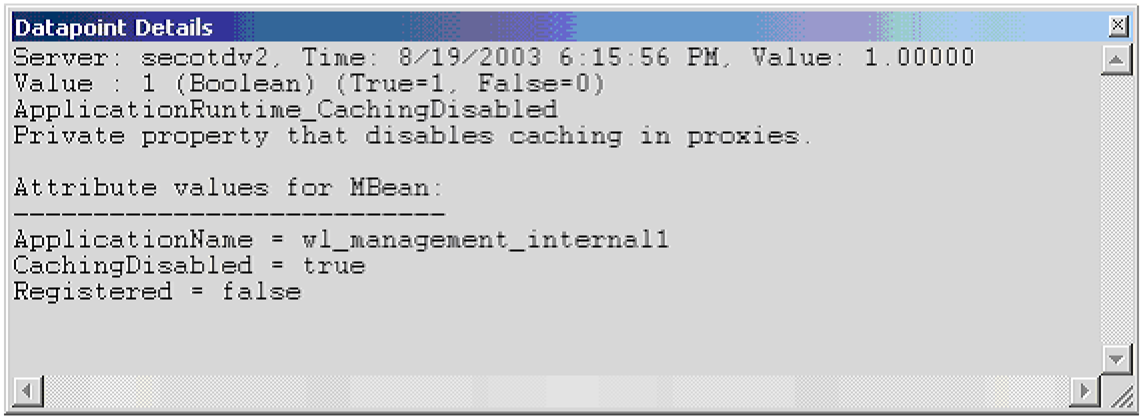
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values and wildcard patterns, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:15:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
ApplicationRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
ApplicationName = wl_management_internal1
CachingDisabled = true
Registered = false
```

Interval

Recommended minimum is 5 minutes.

WebLogic Connector Service Runtime

The WebLogic Connector Service Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
ConnectionPoolCurrentCount	Returns the number of currently deployed connection pools.	Integer	Yes	Yes
ConnectionPoolsTotalCount	Returns the total number of deployed connection pools instantiated since the	Integer	Yes	Yes

	Server startup.			
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Service

Name of the connector service runtime MBean. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter applies only to the counters that are returning a count or total (ConnectionPoolCurrentCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

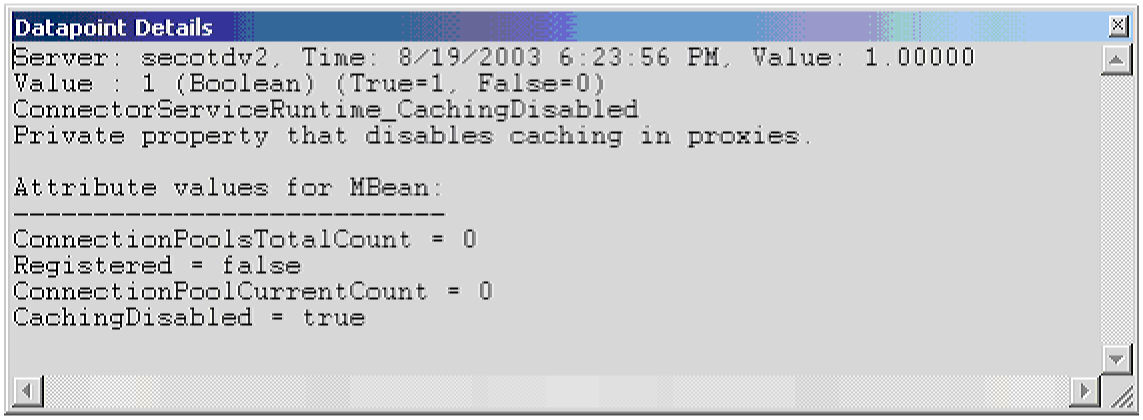
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:23:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
ConnectorServiceRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
ConnectionPoolsTotalCount = 0
Registered = false
ConnectionPoolCurrentCount = 0
CachingDisabled = true
```

Interval

Recommended minimum is 5 minutes.

WebLogic Deployer Runtime

The WebLogic Deployer Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Deployer

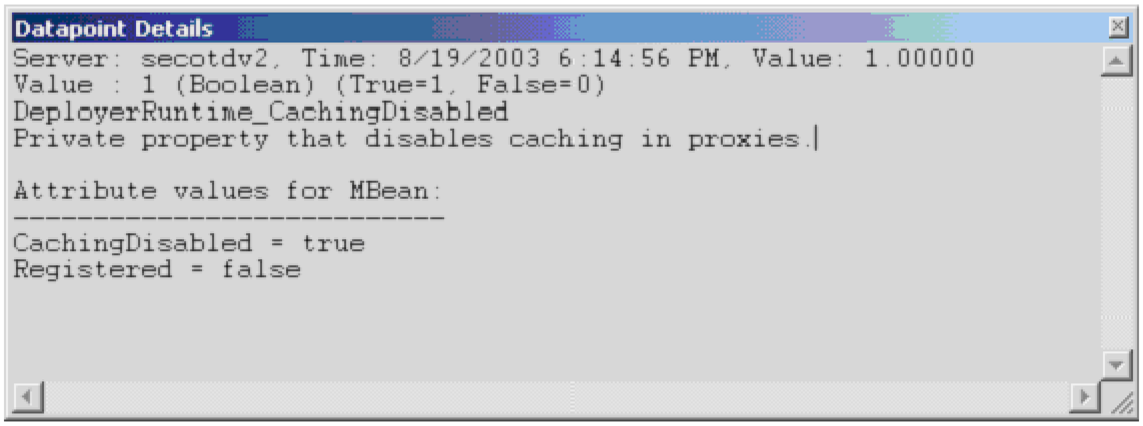
Name of the deployer runtime MBean. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:14:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
DeployerRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
CachingDisabled = true
Registered = false
```

Interval

Recommended minimum is 5 minutes.

WebLogic Domain Log Handler Runtime

The WebLogic Domain Log Handler Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Name

The name of the domain log handler to be monitored. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

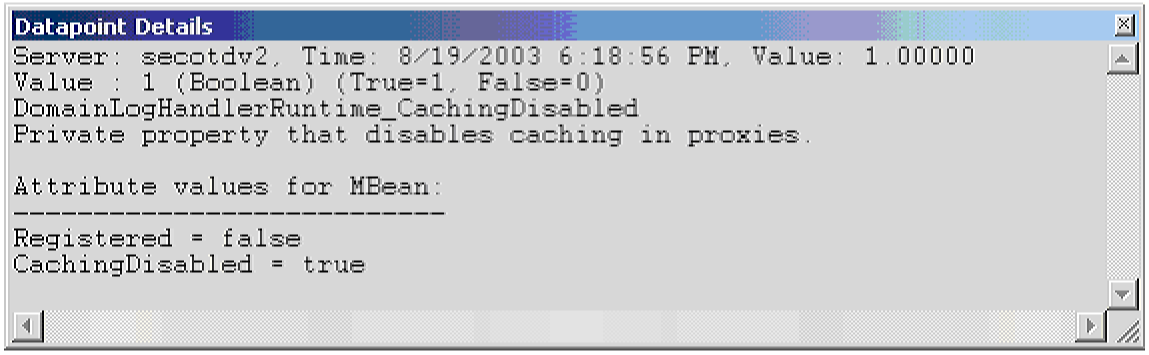
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:18:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
DomainLogHandlerRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
Registered = false
CachingDisabled = true
```

Interval

Recommended minimum is 5 minutes.

WebLogic Domain Runtime

The WebLogic Domain Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x

CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
CurrentClusterDeploymentTimeout	Sets the timeout value in milliseconds of the current deployment to a cluster. This is set at the beginning of the deployment to a cluster and is reset after the deployment.	Long	No	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

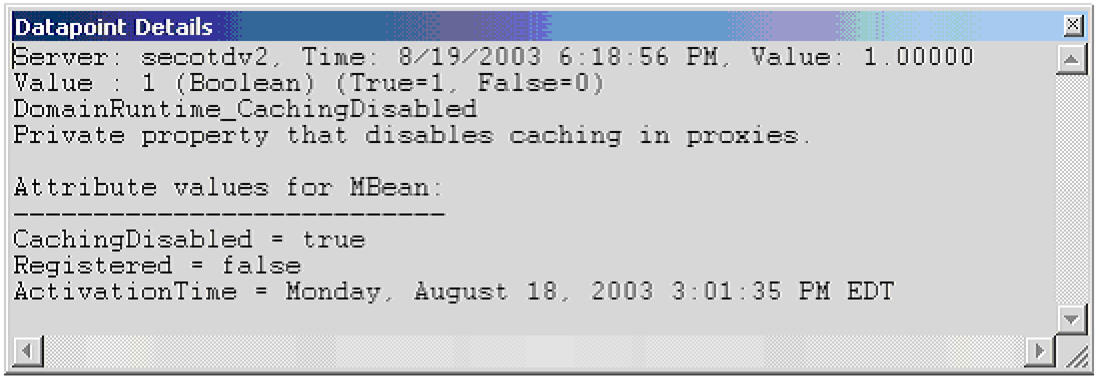
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:18:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
DomainRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
CachingDisabled = true
Registered = false
ActivationTime = Monday, August 18, 2003 3:01:35 PM EDT
```

Interval

Recommended minimum is 5 minutes.

WebLogic EJB Cache Runtime

The WebLogic EJB Cache Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
ActivationCount	Returns the total number of times the EJB was activated.	Long	Yes	Yes
CacheAccessCount	Returns the total number of times EJB was accessed in the cache.	Long	Yes	Yes
CachedBeansCurrentCount	Returns the current number of cached EJBs.	Integer	Yes	Yes
CacheHitCount	Returns the total number of times the EJB was hit in the cache.	Long	Yes	Yes
CacheMissCount	Returns the total number of times an attempt to access a bean from the cache failed.	Long	No	Yes
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
PassivationCount	Returns the total number of times the EJB was passivated.	Long	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Application

The application prefix of the EJB ear. You can specify one or more application prefixes for monitoring. In any combination, select values from the discovered list, or enter values manually.

Component

The EJB component prefix of the EJB ear. You can specify one or more component prefixes for monitoring. In any combination, select values from the discovered list, or enter values manually.

Name

The remainder of the EJB name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (ActivationCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

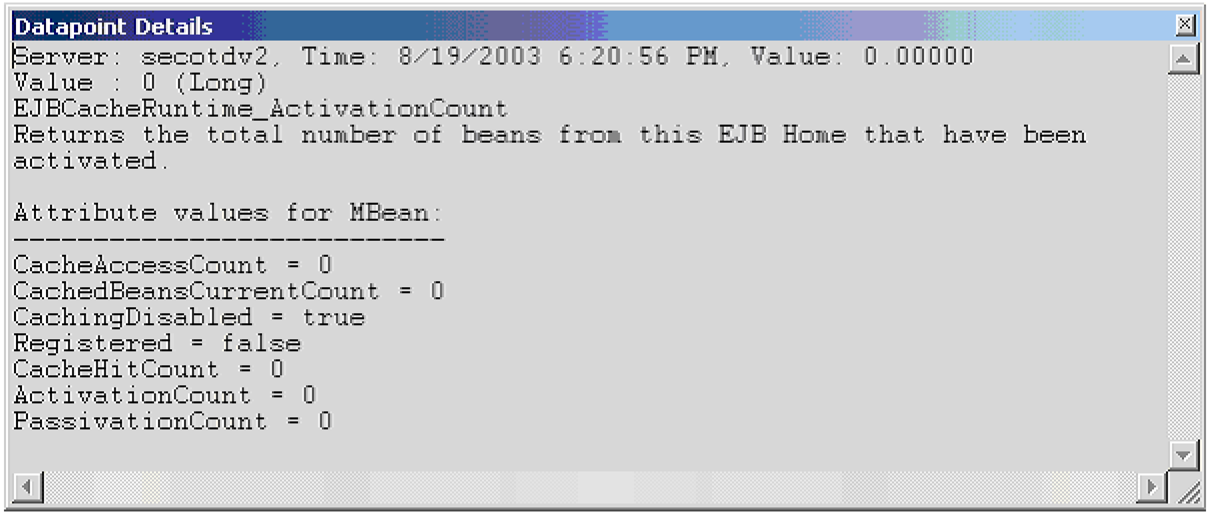
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.


Datapoint Details

```
Server: secotdv2, Time: 8/19/2003 6:20:56 PM, Value: 0.00000
Value : 0 (Long)
EJBCacheRuntime_ActivationCount
Returns the total number of beans from this EJB Home that have been
activated.

Attribute values for MBean:
---------------------------
CacheAccessCount = 0
CachedBeansCurrentCount = 0
CachingDisabled = true
Registered = false
CacheHitCount = 0
ActivationCount = 0
PassivationCount = 0
```

Interval

Recommended minimum is 5 minutes.

WebLogic EJB Component Runtime

The WebLogic EJB Component Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
DeploymentState	Returns current deployment state of the module.	Integer	No	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes
Status	Returns the deployment's status. The set of status is defined in the EJB Deployment interface (DEPLOYED, UNDEPLOYED, ERROR).	Integer	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Application

The application prefix of the EJB ear. You can specify one or more application prefixes for monitoring. In any combination, select values from the discovered list, or enter values manually.

EJBName

The remainder of the EJB component name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

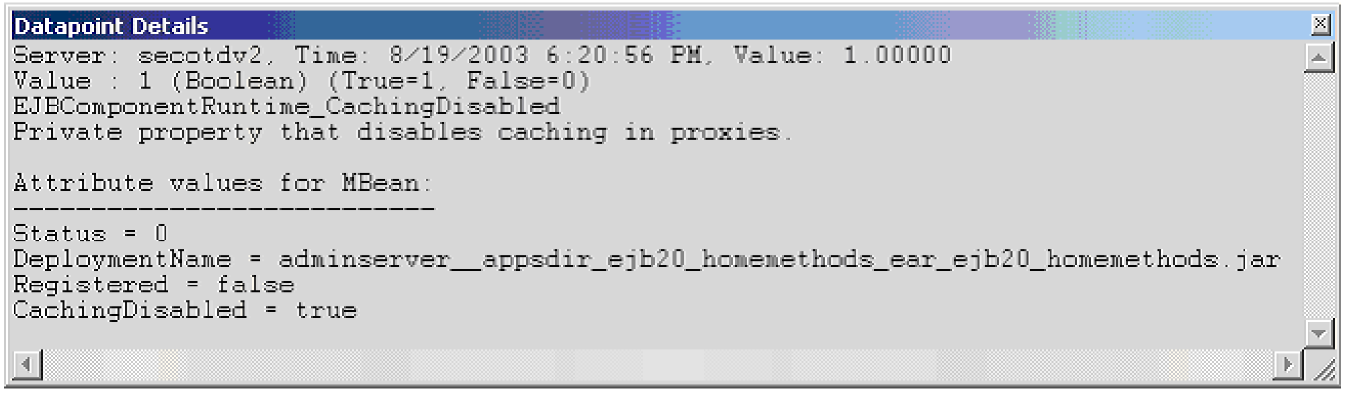
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:20:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
EJBComponentRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
----------------------------
Status = 0
DeploymentName = adminserver__appsdir_ejb20_homemethods_ear_ejb20_homemethods.jar
Registered = false
CachingDisabled = true
```

Interval

Recommended minimum is 5 minutes.

WebLogic EJB Locking Runtime

The WebLogic EJB Locking Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	No
LockEntriesCurrentCount	Returns the number of current EJB lock entries.	Integer	Yes	No
LockManagerAccessCount	Returns the number of accesses to the lock manager.	Long	Yes	No
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	No
TimeoutTotalCount	Returns the number of objects timed out while waiting on the lock.	Long	Yes	No

WaiterTotalCount	Returns the number of objects waiting on the lock.	Long	Yes	No

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Application

The application prefix of the EJB ear. You can specify one or more application prefixes for monitoring. In any combination, select values from the discovered list, or enter values manually.

Component

The EJB component prefix of the EJB ear. You can specify one or more component prefixes for monitoring. In any combination, select values from the discovered list, or enter values manually.

Name

The remainder of the EJB name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (LockEntriesCurrentCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

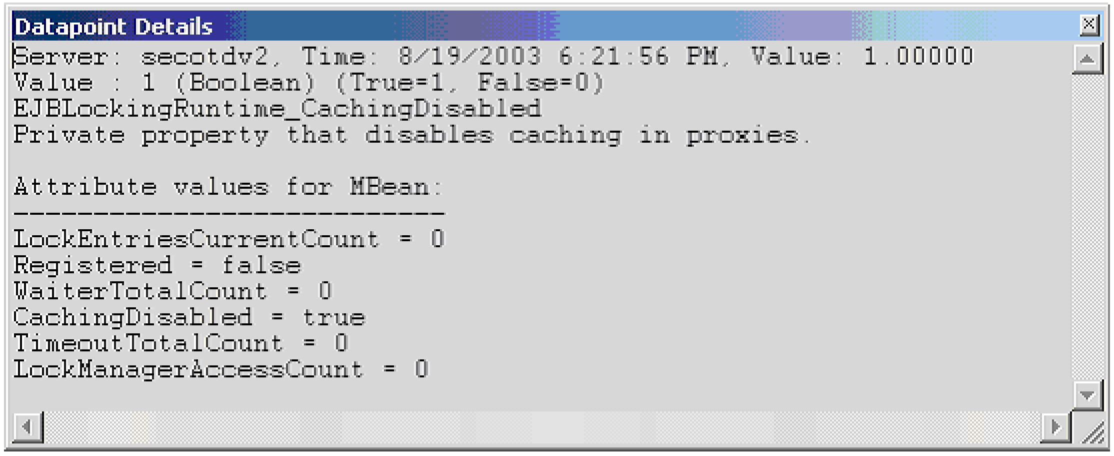
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:21:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
EJBLockingRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
LockEntriesCurrentCount = 0
Registered = false
WaiterTotalCount = 0
CachingDisabled = true
TimeoutTotalCount = 0
LockManagerAccessCount = 0
```

Interval

Recommended minimum is 5 minutes.

WebLogic EJB Pool Runtime

The WebLogic EJB Pool Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
AccessTotalCount	Returns the total number of times an attempt was made to get an instance from the free pool.	Long	No	Yes
BeansInUseCount	Returns the number of beans currently in use.	Integer	Yes	Yes
BeansInUseCurrentCount	Returns the number of bean instances currently in use from the free pool.	Integer	No	Yes
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
DestroyedTotalCount	Returns the total number of times a bean instance from this pool was destroyed due to a non-application Exception being thrown from it.	Long	No	Yes
IdleBeansCount	Returns the number of idle beans in this EJB.	Integer	Yes	Yes
MissTotalCount	Returns the total number of times	Long	No	Yes

	a failed attempt was made to get an instance from the free pool. An attempt to get a bean from the pool fails if there are no available instances in the pool.			
PooledBeansCurrentCount	Returns the current number of available bean instances in the free pool.	Integer	No	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes
TimeoutTotalCount	Returns the total number of timed out transactions.	Long	Yes	Yes
WaiterCurrentCount	Returns the current number of available bean instances in the free pool.	Integer	No	Yes
WaiterTotalCount	Returns the number of EJBs currently waiting.	Long	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Application

The application prefix of the EJB ear. You can specify one or more application prefixes for monitoring. In any combination, select values from the discovered list, or enter values manually.

Component

The EJB component prefix of the EJB ear. You can specify one or more component prefixes for monitoring. In any combination, select values from the discovered list, or enter values manually.

Name

The remainder of the EJB name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (BeansInUseCount is one example in this counter category). Possible values are:

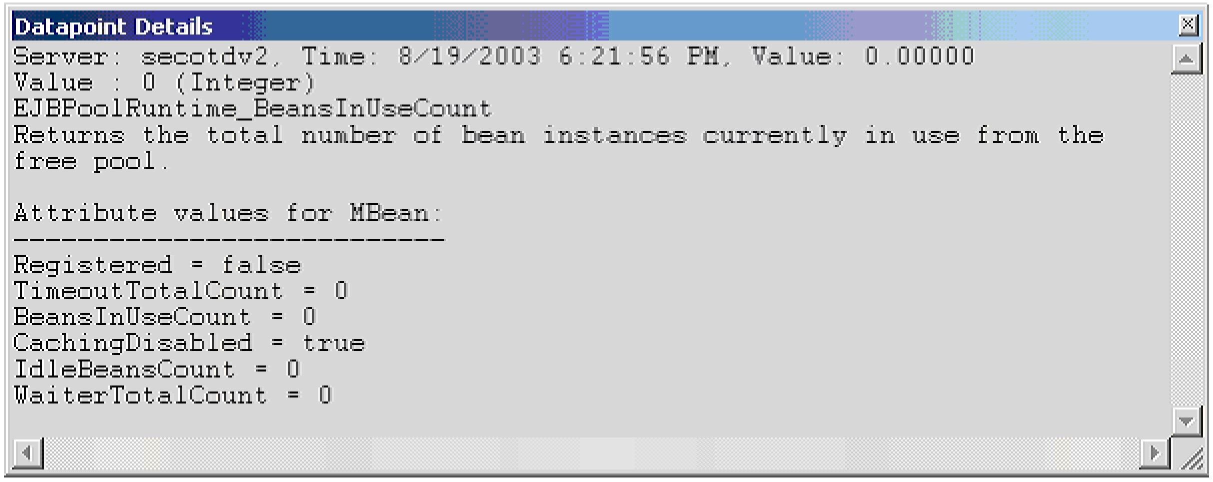
ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:21:56 PM, Value: 0.00000
Value : 0 (Integer)
EJBPoolRuntime_BeansInUseCount
Returns the total number of bean instances currently in use from the
free pool.

Attribute values for MBean:
---------------------------
Registered = false
TimeoutTotalCount = 0
BeansInUseCount = 0
CachingDisabled = true
IdleBeansCount = 0
WaiterTotalCount = 0
```

Interval

Recommended minimum is 5 minutes.

WebLogic EJB Transaction Runtime

The WebLogic EJB Transaction Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes

Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes
TransactionsCommittedTotalCount	Returns the total number of EJB transactions that were committed.	Long	Yes	Yes
TransactionsRolledBackTotalCount	Returns the total number of EJB transactions rolled back.	Long	Yes	Yes
TransactionsTimedOutTotalCount	Returns the total number of EJB transactions that timed out.	Long	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Application

The application prefix of the EJB ear. You can specify one or more application prefixes for monitoring. In any combination, select values from the discovered list, or enter values manually.

Component

The EJB component prefix of the EJB ear. You can specify one or more component prefixes for monitoring. In any combination, select values from the discovered list, or enter values manually.

Name

The remainder of the EJB name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (TransactionsCommittedTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last

task interval and the raw data value of the counter in the current task interval.

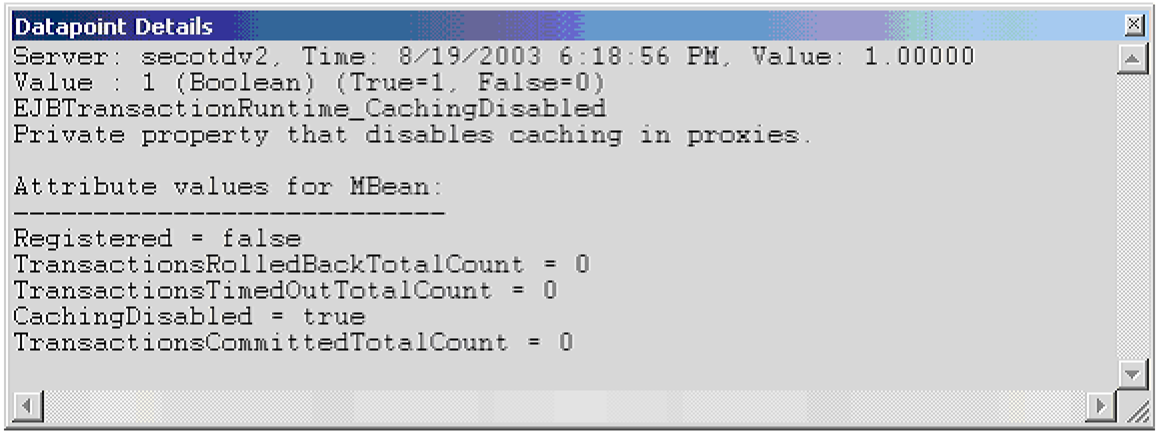
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.


Datapoint Details

```
Server: secotdv2, Time: 8/19/2003 6:18:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
EJBTransactionRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
Registered = false
TransactionsRolledBackTotalCount = 0
TransactionsTimedOutTotalCount = 0
CachingDisabled = true
TransactionsCommittedTotalCount = 0
```

Interval

Recommended minimum is 5 minutes.

WebLogic Entity EJB Runtime

The WebLogic Entity EJB Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Application

The application prefix of the EJB ear. You can specify one or more application prefixes for monitoring. In any combination, select values from the discovered list, or enter values manually.

Component

The EJB component prefix of the EJB ear. You can specify one or more component prefixes for monitoring. In any combination, select values from the discovered list, or enter values manually.

Name

The remainder of the EJB name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

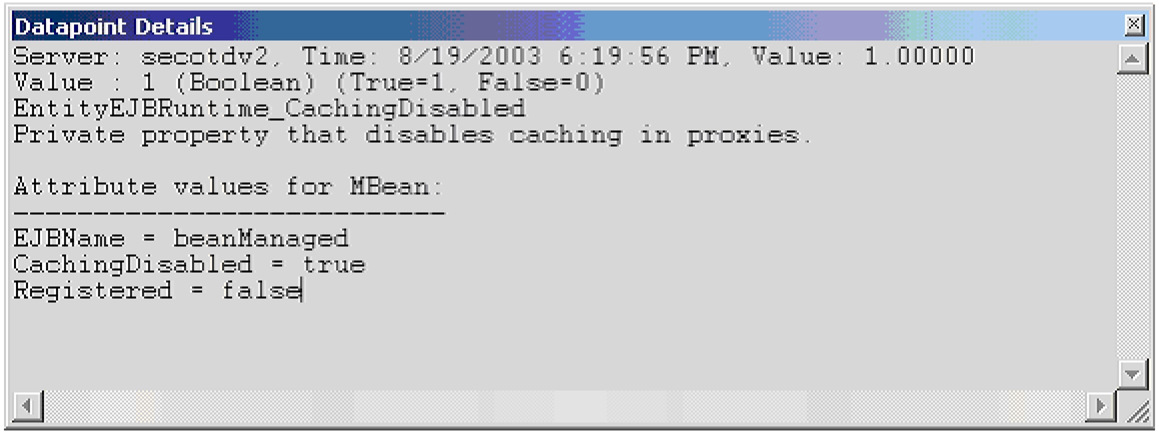
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:19:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
EntityEJBRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
EJBName = beanManaged
CachingDisabled = true
Registered = false
```

Interval

Recommended minimum is 5 minutes.

WebLogic Execute Queue Runtime

The WebLogic Execute Queue Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
ExecuteThreadCurrentIdleCount	Returns the number of idle threads assigned to the queue.	Integer	Yes	Yes
ExecuteThreadTotalCount	Returns the total number of execute threads assigned to the queue.	Integer	No	Yes
PendingRequestCurrentCount	Returns the number of waiting requests in the queue.	Integer	Yes	Yes
PendingRequestOldestTime	Returns the time that the longest waiting request was placed in the queue.	Long	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes
ServicedRequestTotalCount	Returns the number of requests that have been processed by this queue.	Integer	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Queue

The execution queue name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (ExecuteThreadTotalCount is one example in this counter category). Possible values are:

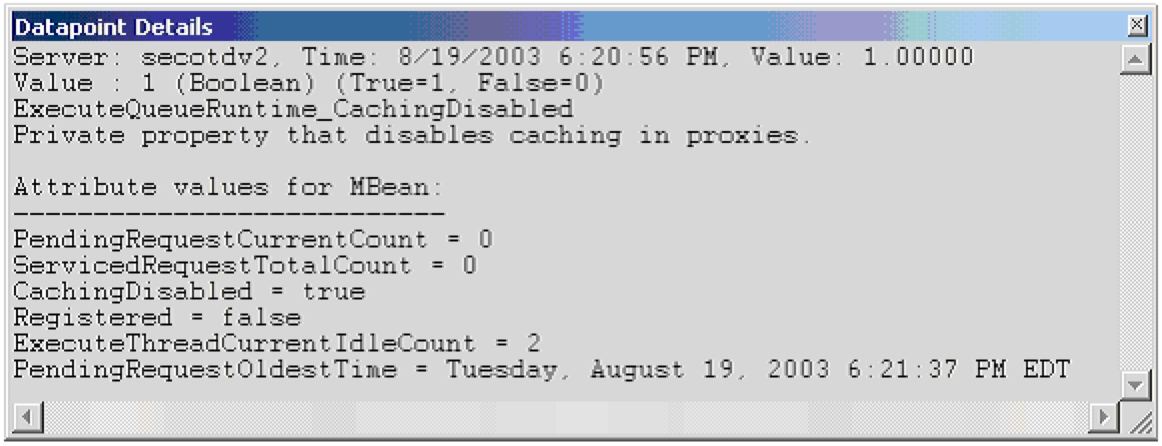
ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:20:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
ExecuteQueueRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
PendingRequestCurrentCount = 0
ServicedRequestTotalCount = 0
CachingDisabled = true
Registered = false
ExecuteThreadCurrentIdleCount = 2
PendingRequestOldestTime = Tuesday, August 19, 2003 6:21:37 PM EDT
```

Interval

Recommended minimum is 5 minutes.

WebLogic JDBC Connection Pool Runtime

The WebLogic JDBC Connection Pool Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
ActiveConnectionsAverageCount	Returns the running average of active connections in the this MBean. The count starts at zero each time the MBean is instantiated.	Integer	No	Yes
ActiveConnectionsCurrentCount	Returns the current number of	Integer	Yes	Yes

	active connections.			
ActiveConnectionsHighCount	Returns the highest number of active current connections. The count starts at zero each time the JDBCConnectionPoolRuntime MBean is instantiated.	Integer	Yes	Yes
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
ConnectionDelayTime	Returns the number of milliseconds it takes to get a physical connection from database. It is calculated as summary time to connect, divided by summary number of connections.	Integer	Yes	Yes
ConnectionLeakProfileCount	Returns the current number of connection leak profiles in the profile storage.	Integer	Yes	Yes
ConnectionsTotalCount	Returns the total number of JDBC connections in this JDBCConnectionPoolRuntime MBean since the pool was instantiated.	Integer	Yes	Yes
FailuresToReconnectCount	Returns the count of attempts to refresh a connection to a database that failed. Failure may happen because of database unavailability or a broken connection to the database.	Integer	Yes	Yes
HighestNumAvailable	Returns the highest number of available connections in this pool.	Integer	No	Yes
HighestNumUnavailable	Returns the highest number of unavailable connections in this pool.	Integer	No	Yes
LeakedConnectionCount	Returns the number of connections that were checked out from the connection pool but were not returned to the pool by calling close ().	Integer	Yes	Yes
MaxCapacity	Returns the maximum capacity of this connection pool.	Integer	Yes	Yes
NumAvailable	Returns the number of available connections in this pool.	Integer	No	Yes
NumUnavailable	Returns the number of unavailable connections in this pool.	Integer	No	Yes

PoolState	Returns true if the pool is enabled, false if the pool is disabled.	Boolean	Yes	Yes
PreparedStatementCacheProfileCount	Returns the number of prepared statement cache profiling stores cache snapshots that are in external storage.	Integer	Yes	Yes
PrepStmtCacheHitCount	Returns the cumulative, running count of the use of each cached statement.	Integer	Yes	Yes
PrepStmtCacheMissCount	Returns a count of the cases when the cache does not have a cached statement to satisfy a request.	Integer	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes
StatementProfileCount	Returns the number of statement profiling stores in external storage.	Integer	Yes	Yes
WaitingForConnectionCurrentCount	Returns the current number of requests waiting for a connection.	Integer	Yes	Yes
WaitingForConnectionHighCount	Returns the highest number of requests waiting for a connection. The count starts at zero each time the JDBCConnectionPoolRuntime MBean is instantiated.	Integer	Yes	Yes
WaitSecondsHighCount	Returns the highest number of seconds a connection waited.	Integer	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Pool

The connection pool name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (ConnectionsTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

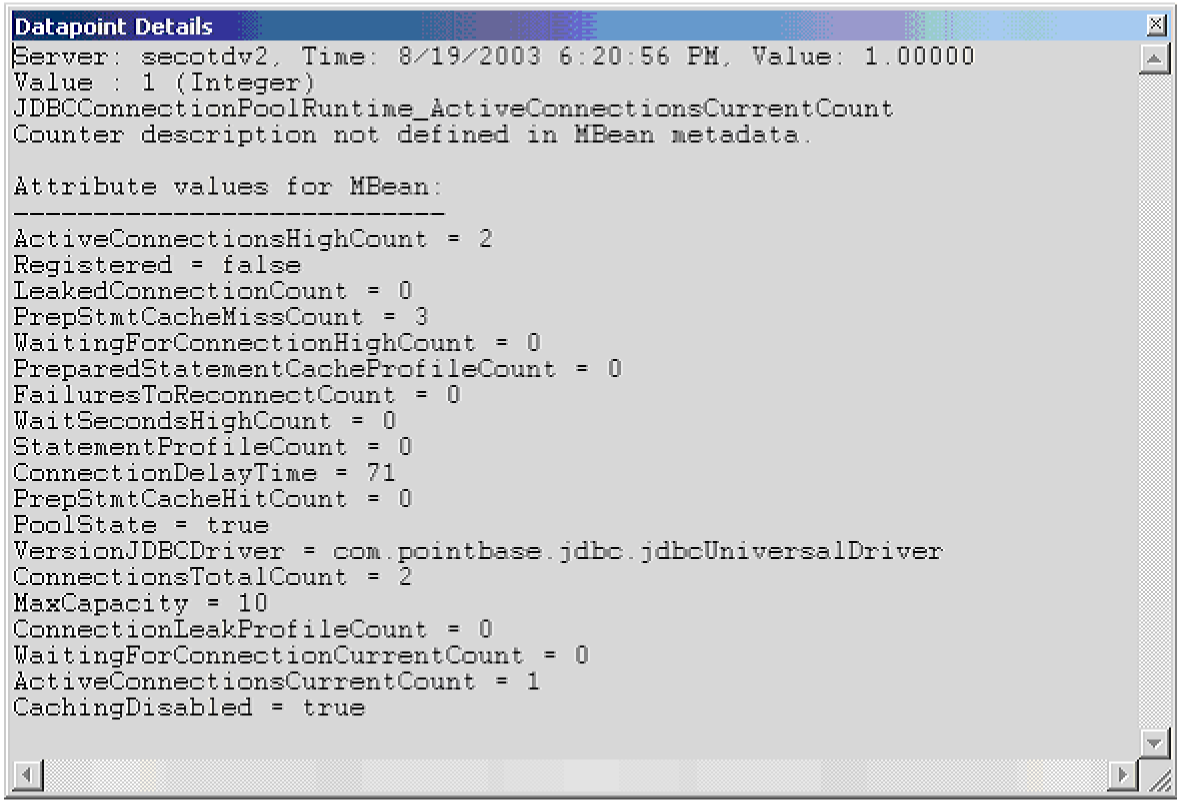
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:20:56 PM, Value: 1.00000
Value : 1 (Integer)
JDBCConnectionPoolRuntime_ActiveConnectionsCurrentCount
Counter description not defined in MBean metadata.

Attribute values for MBean:
---------------------------
ActiveConnectionsHighCount = 2
Registered = false
LeakedConnectionCount = 0
PrepStmtCacheMissCount = 3
WaitingForConnectionHighCount = 0
PreparedStatementCacheProfileCount = 0
FailuresToReconnectCount = 0
WaitSecondsHighCount = 0
StatementProfileCount = 0
ConnectionDelayTime = 71
PrepStmtCacheHitCount = 0
PoolState = true
VersionJDBCDriver = com.pointbase.jdbc.jdbcUniversalDriver
ConnectionsTotalCount = 2
MaxCapacity = 10
ConnectionLeakProfileCount = 0
WaitingForConnectionCurrentCount = 0
ActiveConnectionsCurrentCount = 1
CachingDisabled = true
```

Interval

Recommended minimum is 5 minutes.

WebLogic JMS Connection Runtime

The WebLogic JMS Connection Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes
SessionsCurrentCount	Returns the current number of sessions for this connection.	Long	Yes	Yes
SessionsHighCount	Returns the peak number of sessions for this connection since the last reset.	Long	Yes	Yes
SessionsTotalCount	Returns the number of sessions on this connection since the last reset.	Long	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Connection

The JMS connection name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (SessionsTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.

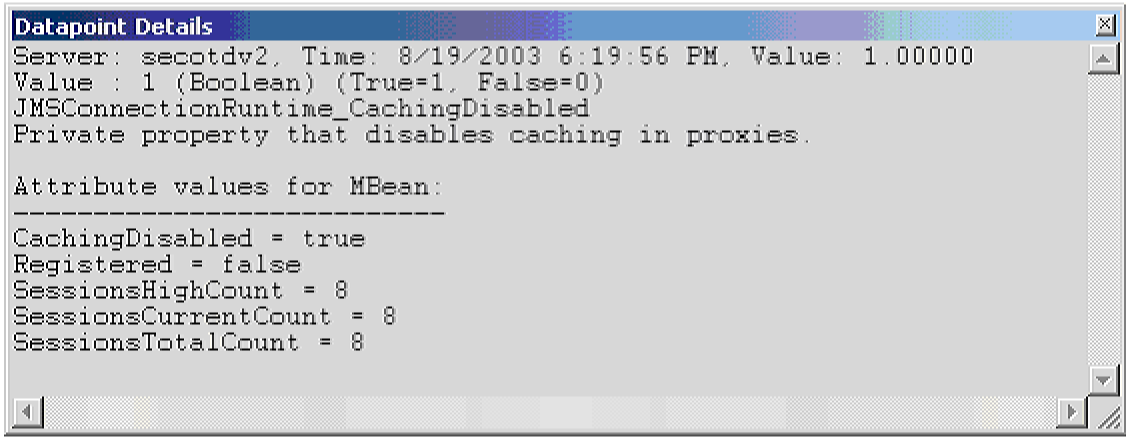
INTERVAL The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:19:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
JMSConnectionRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
CachingDisabled = true
Registered = false
SessionsHighCount = 8
SessionsCurrentCount = 8
SessionsTotalCount = 8
```

Interval

Recommended minimum is 5 minutes.

WebLogic JMS Consumer Runtime

The WebLogic JMS Consumer Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
Active	Determines if the consumer is active. Determines whether the consumer has a message listener set up or a synchronous receive in progress.	Boolean	Yes	Yes
BytesPendingCount	Returns the number of bytes pending (uncommitted and unacknowledged) by this consumer.	Long	Yes	Yes
BytesReceivedCount	Returns the number of bytes received by this consumer since the last reset.	Long	Yes	Yes
CachingDisabled	Private property that disables caching in	Boolean	Yes	Yes

	proxies.			
Durable	Determines whether the consumer is durable.	Boolean	Yes	Yes
MessagesPendingCount	Returns the number of messages pending (uncommitted and unacknowledged) by this consumer.	Long	Yes	Yes
MessagesReceivedCount	Returns the number of messages received by this consumer since the last reset.	Long	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Consumer

The JMS consumer name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (MessagesReceivedCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

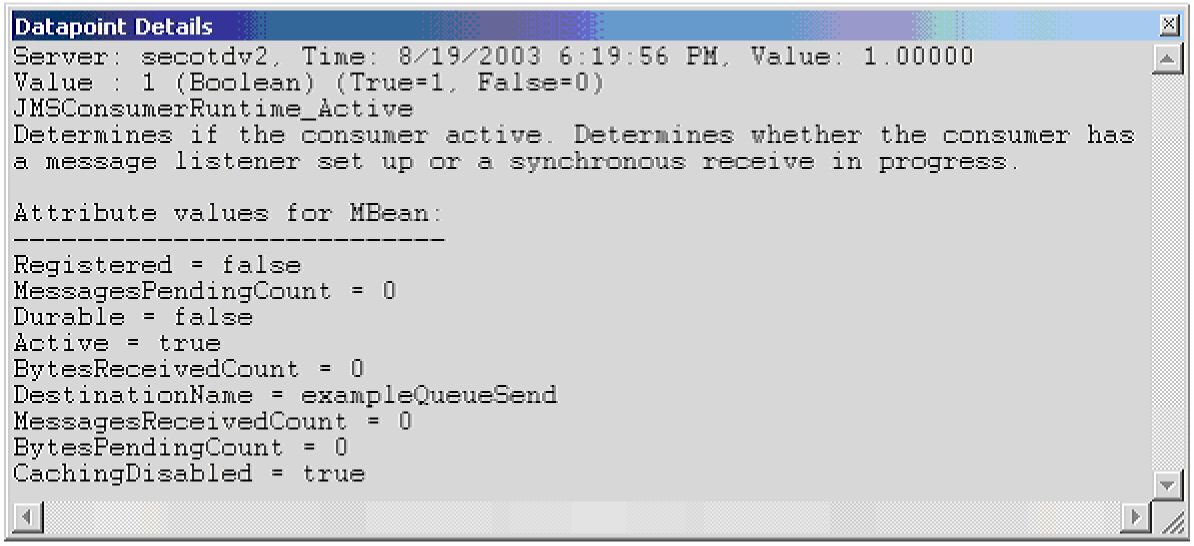
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:19:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
JMSConsumerRuntime_Active
Determines if the consumer active. Determines whether the consumer has
a message listener set up or a synchronous receive in progress.

Attribute values for MBean:
---------------------------
Registered = false
MessagesPendingCount = 0
Durable = false
Active = true
BytesReceivedCount = 0
DestinationName = exampleQueueSend
MessagesReceivedCount = 0
BytesPendingCount = 0
CachingDisabled = true
```

Interval

Recommended minimum is 5 minutes.

WebLogic JMS Destination Runtime

The WebLogic JMS Destination Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
BytesCurrentCount	Returns the current number of bytes stored in the destination. This does not include the pending bytes.	Long	Yes	Yes
BytesHighCount	Returns the peak number of bytes stored in the destination since the last reset.	Long	Yes	Yes
BytesPendingCount	Returns the number of pending bytes stored in the destination. Pending bytes are over and above the current number of bytes.	Long	Yes	Yes
BytesReceivedCount	Returns the number of bytes received in this destination since the last reset.	Long	Yes	Yes
BytesThresholdTime	Returns the amount of time in the threshold condition since the last reset.	Long	Yes	Yes

CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
ConsumersCurrentCount	Returns the current number of consumers accessing this destination.	Long	Yes	Yes
ConsumersHighCount	Returns the peak number of consumers accessing this destination since the last reset.	Long	Yes	Yes
ConsumersTotalCount	Returns the total number of consumers accessing this destination since the last reset.	Long	Yes	Yes
MessagesCurrentCount	Returns the current number of messages in the destination. This does not include the pending messages.	Long	Yes	Yes
MessagesHighCount	Returns the peak number of messages in the destination since the last reset.	Long	Yes	Yes
MessagesPendingCount	Returns the number of pending messages in the destination. Pending messages are over and above the current number of messages. A pending message is one that has either been sent in a transaction and not committed, or that has been received and not committed or acknowledged.	Long	Yes	Yes
MessagesReceivedCount	Returns the number of messages received in this destination since that reset.	Long	Yes	Yes
MessagesThresholdTime	Returns the amount of time in the threshold condition since the last reset.	Long	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Destination

The name of the JMS destination. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (MessagesReceivedCount is one example in this counter category). Possible values are:

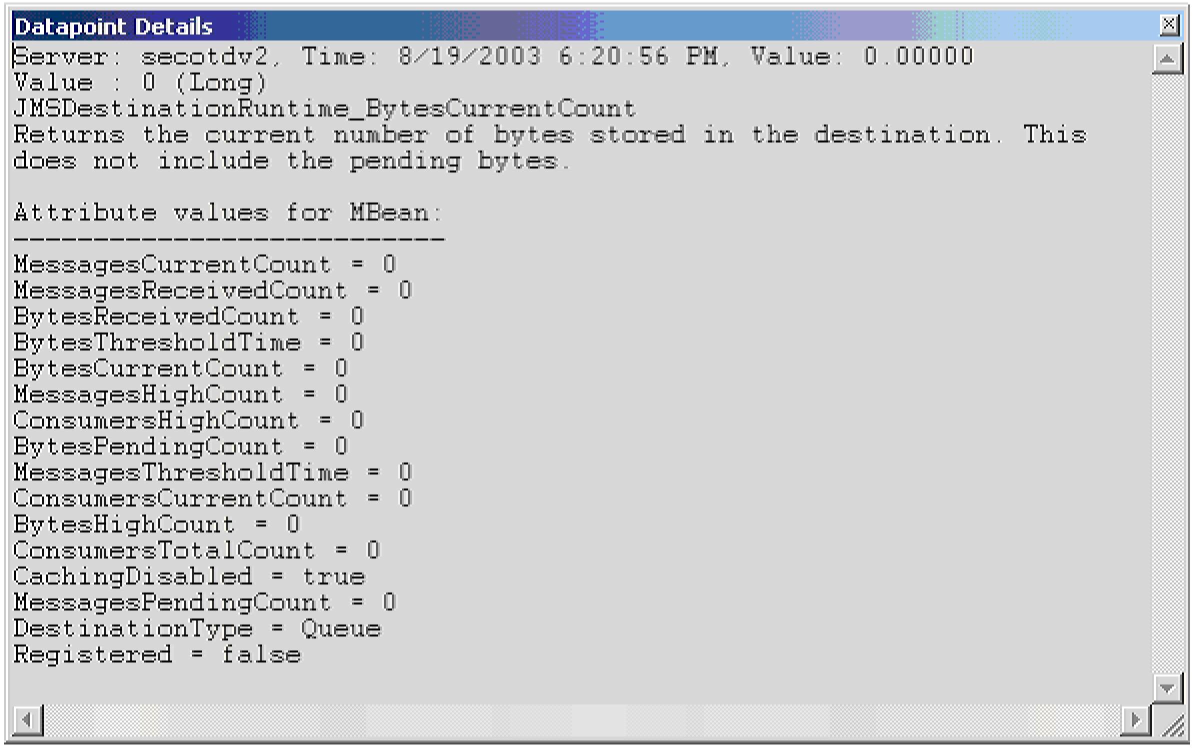
ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:20:56 PM, Value: 0.00000
Value : 0 (Long)
JMSDestinationRuntime_BytesCurrentCount
Returns the current number of bytes stored in the destination. This
does not include the pending bytes.

Attribute values for MBean:
---------------------------
MessagesCurrentCount = 0
MessagesReceivedCount = 0
BytesReceivedCount = 0
BytesThresholdTime = 0
BytesCurrentCount = 0
MessagesHighCount = 0
ConsumersHighCount = 0
BytesPendingCount = 0
MessagesThresholdTime = 0
ConsumersCurrentCount = 0
BytesHighCount = 0
ConsumersTotalCount = 0
CachingDisabled = true
MessagesPendingCount = 0
DestinationType = Queue
Registered = false
```

Interval

Recommended minimum is 5 minutes.

WebLogic JMS Runtime

The WebLogic JMS Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
ConnectionsCurrentCount	Returns the current number of connections to this WebLogic Server.	Long	Yes	Yes
ConnectionsHighCount	Returns the peak number of connections to this WebLogic Server since the last reset.	Long	Yes	Yes
ConnectionsTotalCount	Returns the total number of connections made to this WebLogic Server since the last reset.	Long	Yes	Yes
JMSServersCurrentCount	Returns the current number of JMS servers that are deployed on this WebLogic Server instance.	Long	Yes	Yes
JMSServersHighCount	Returns the peak number of JMS servers that were deployed on this WebLogic Server instance since the server was started.	Long	Yes	Yes
JMSServersTotalCount	Returns the number of JMS servers that were deployed on this WebLogic Server instance since the server was started.	Long	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

JMSServer

The JMS server name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (ConnectionsTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

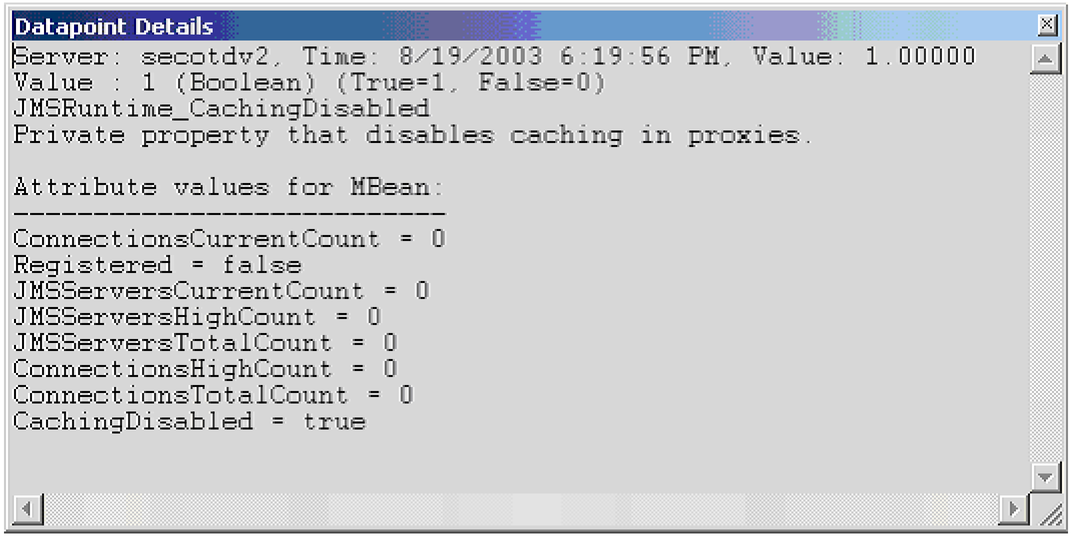
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:19:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
JMSRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
ConnectionsCurrentCount = 0
Registered = false
JMSServersCurrentCount = 0
JMSServersHighCount = 0
JMSServersTotalCount = 0
ConnectionsHighCount = 0
ConnectionsTotalCount = 0
CachingDisabled = true
```

Interval

Recommended minimum is 5 minutes.

WebLogic JMS Server Runtime

The WebLogic JMS Server Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX

Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
BytesCurrentCount	Returns the current number of bytes stored on this JMS server. This does not include the pending bytes.	Long	Yes	Yes
BytesHighCount	Returns the peak number of bytes stored in the JMS server since the last reset.	Long	Yes	Yes
BytesPendingCount	Returns the current number of bytes pending (unacknowledged or uncommitted) stored on this JMS server. Pending bytes are over and above the current number of bytes.	Long	Yes	Yes
BytesReceivedCount	Returns the number of bytes received on this JMS server since the last reset.	Long	Yes	Yes
BytesThresholdTime	Returns the amount of time in the threshold condition since the last reset.	Long	Yes	Yes
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
DestinationsCurrentCount	Returns the current number of destinations for this JMS server.	Long	Yes	Yes
DestinationsHighCount	Returns the peak number of destinations on this JMS server since the last reset.	Long	Yes	Yes
DestinationsTotalCount	Returns the number of destinations instantiated on this JMS server since the last reset.	Long	Yes	Yes
MessagesCurrentCount	Returns the current number of messages stored on this JMS server. This does not include the pending messages.	Long	Yes	Yes
MessagesHighCount	Returns the peak number of messages stored in the JMS server since the last reset.	Long	Yes	Yes
MessagesPendingCount	Returns the current number of messages pending (unacknowledged or uncommitted) stored on this JMS server. Pending messages are over and above the current number of messages.	Long	Yes	Yes
MessagesReceivedCount	Returns the number of messages received on this destination since the last reset.	Long	Yes	Yes
MessagesThresholdTime	Returns the amount of time in the threshold condition since the last reset.	Long	Yes	Yes

Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes
SessionPoolsCurrentCount	Returns the current number of session pools instantiated on this JMS server.	Long	Yes	Yes
SessionPoolsHighCount	Returns the peak number of session pools instantiated on this JMS server since the last reset.	Long	Yes	Yes
SessionPoolsTotalCount	Returns the number of session pools instantiated on this JMS server since the last reset.	Long	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

JMSServer

The JMS server name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (SessionPoolsTotalCount is one example in this counter category). Possible values are:

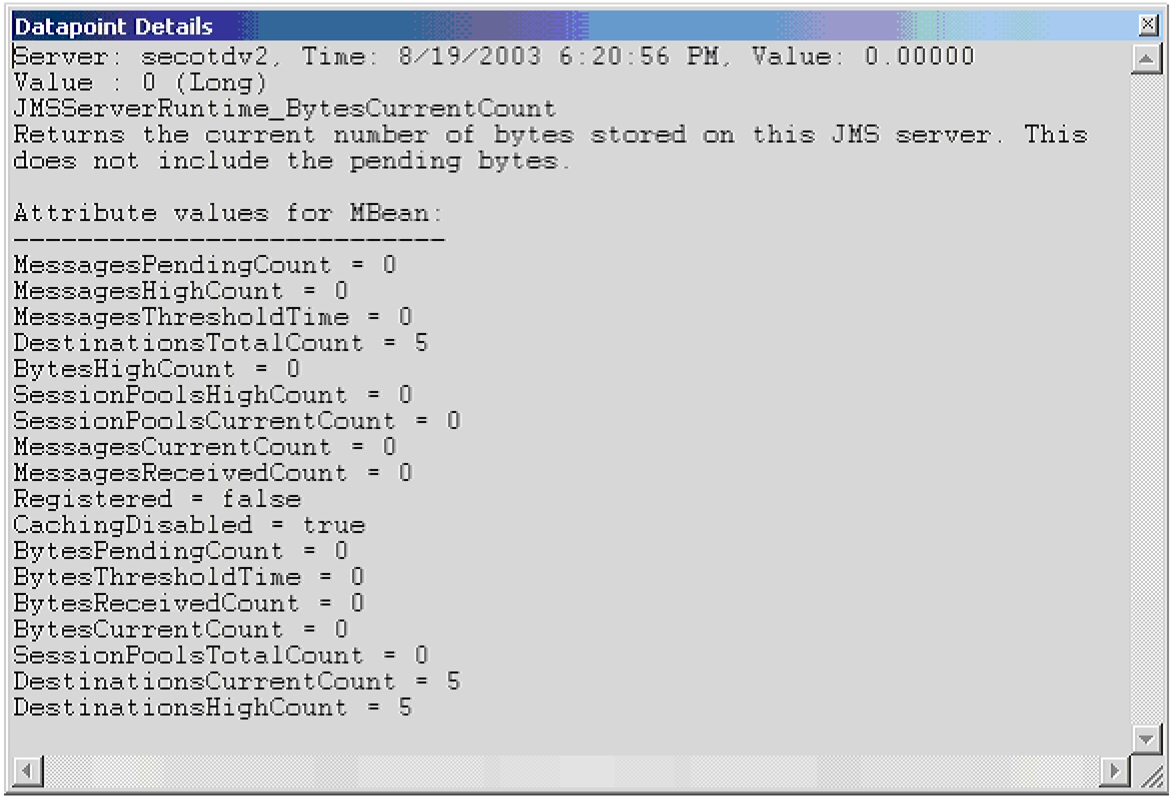
ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:20:56 PM, Value: 0.00000
Value : 0 (Long)
JMSServerRuntime_BytesCurrentCount
Returns the current number of bytes stored on this JMS server. This
does not include the pending bytes.

Attribute values for MBean:
---------------------------
MessagesPendingCount = 0
MessagesHighCount = 0
MessagesThresholdTime = 0
DestinationsTotalCount = 5
BytesHighCount = 0
SessionPoolsHighCount = 0
SessionPoolsCurrentCount = 0
MessagesCurrentCount = 0
MessagesReceivedCount = 0
Registered = false
CachingDisabled = true
BytesPendingCount = 0
BytesThresholdTime = 0
BytesReceivedCount = 0
BytesCurrentCount = 0
SessionPoolsTotalCount = 0
DestinationsCurrentCount = 5
DestinationsHighCount = 5
```

Interval

Recommended minimum is 5 minutes.

WebLogic JMS Session Runtime

The WebLogic JMS Session Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
BytesPendingCount	Returns the number of bytes pending (uncommitted and unacknowledged) for this session.	Long	Yes	Yes
BytesReceivedCount	Returns the number of bytes received by this session since the last reset.	Long	Yes	Yes
BytesSentCount	Returns the number of bytes sent by this session since the last reset.	Long	Yes	Yes
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes

ConsumersCurrentCount	Returns the current number of consumers for this session.	Long	Yes	Yes
ConsumersHighCount	Returns the peak number of consumers for this session since the last reset.	Long	Yes	Yes
ConsumersTotalCount	Returns the number of consumers instantiated by this session since the last reset.	Long	Yes	Yes
MessagesPendingCount	Returns the number of messages pending (uncommitted and unacknowledged) for this session.	Long	Yes	Yes
MessagesReceivedCount	Returns the number of messages sent by this session since the last reset.	Long	Yes	Yes
MessagesSentCount	Returns the number of bytes sent by this session since the last reset.	Long	Yes	Yes
ProducersCurrentCount	Returns the current number of producers for this session.	Long	Yes	Yes
ProducersHighCount	Returns the peak number of producers for this session since the last reset.	Long	Yes	Yes
ProducersTotalCount	Returns the number of producers for this session since the last reset.	Long	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes
Transacted	Returns whether the session is transacted.	Boolean	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Session

The JMS session name. You can select a value from the discovered list.

StatType

This parameter is available for counters that are returning a count or total (ConsumersTotalCount is one example in this counter category). Possible values are:

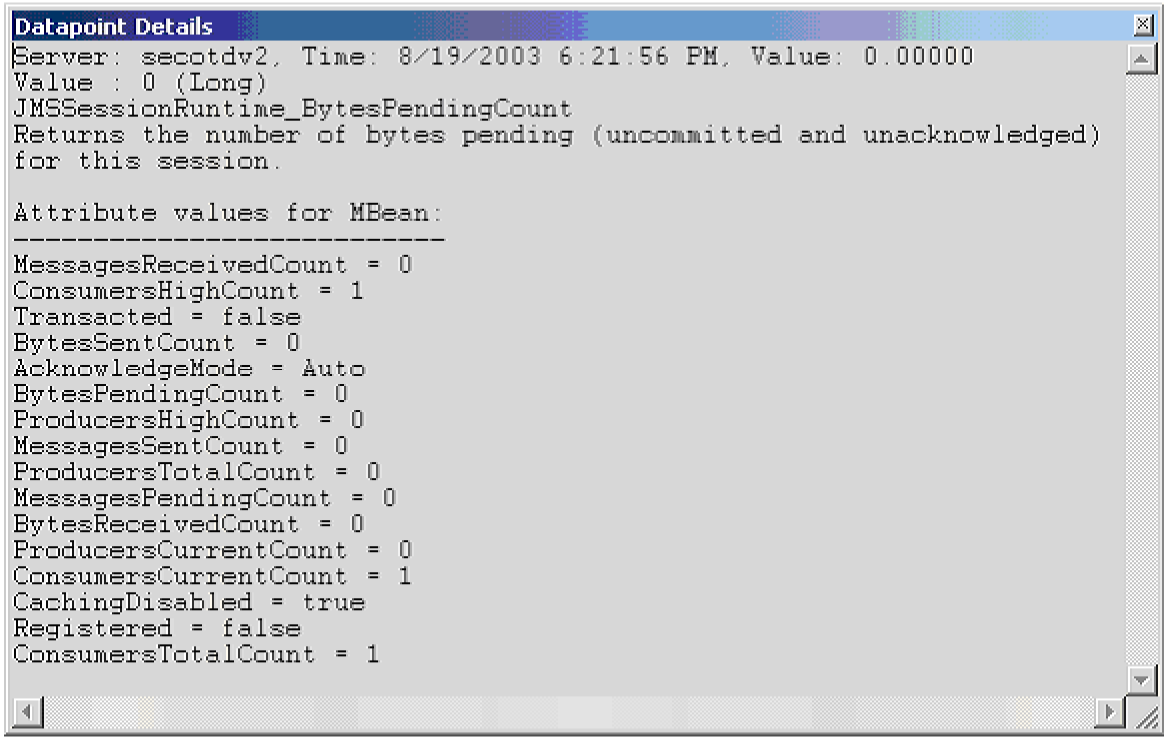
ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:21:56 PM, Value: 0.00000
Value : 0 (Long)
JMSSessionRuntime_BytesPendingCount
Returns the number of bytes pending (uncommitted and unacknowledged)
for this session.

Attribute values for MBean:
---------------------------
MessagesReceivedCount = 0
ConsumersHighCount = 1
Transacted = false
BytesSentCount = 0
AcknowledgeMode = Auto
BytesPendingCount = 0
ProducersHighCount = 0
MessagesSentCount = 0
ProducersTotalCount = 0
MessagesPendingCount = 0
BytesReceivedCount = 0
ProducersCurrentCount = 0
ConsumersCurrentCount = 1
CachingDisabled = true
Registered = false
ConsumersTotalCount = 1
```

Interval

Recommended minimum is 5 minutes.

WebLogic JTA Recovery Runtime

The WebLogic JTA Recovery Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX

Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
Active	Returns whether the Transaction Recovery Service is currently activated on this server.	Boolean	Yes	Yes
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
InitialRecoveredTransactionTotalCount	Returns the total number of transactions that are recovered from the Transaction Log initially.	Integer	Yes	Yes
RecoveredTransactionCompletionPercent	Returns the percentage of the initially recovered transactions that are completed.	Integer	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (InitialRecoveredTransactionTotalCount is one example in this counter category). Possible values are:

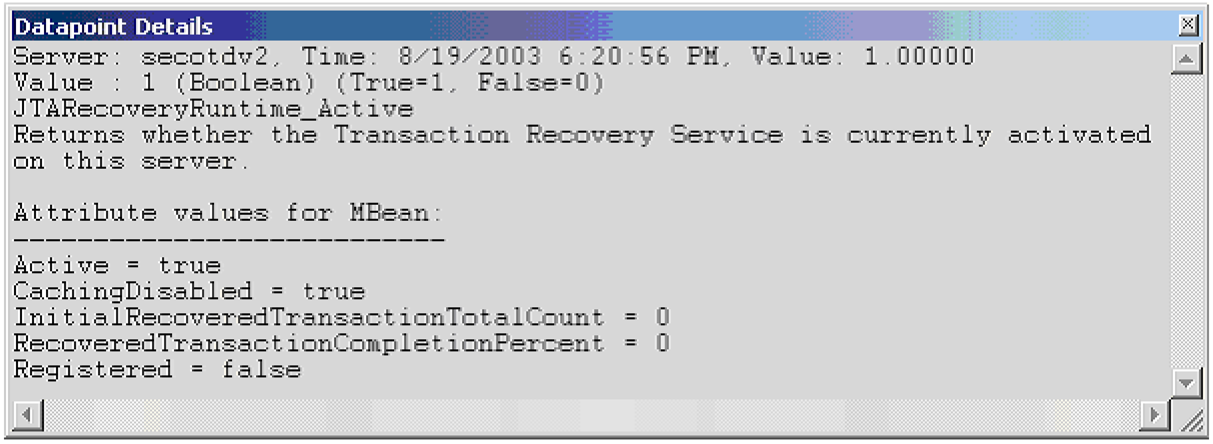
ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:20:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
JTARecoveryRuntime_Active
Returns whether the Transaction Recovery Service is currently activated
on this server.

Attribute values for MBean:
---------------------------
Active = true
CachingDisabled = true
InitialRecoveredTransactionTotalCount = 0
RecoveredTransactionCompletionPercent = 0
Registered = false
```

Interval

Recommended minimum is 5 minutes.

WebLogic JTA Runtime

The WebLogic JTA Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
ActiveTransactionsTotalCount	Returns the number of active transactions on the server.	Integer	Yes	Yes
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes
SecondsActiveTotalCount	Returns the total number of seconds for all committed transactions	Long	Yes	Yes
TransactionAbandonedTotalCount	Returns the number of transactions that were	Long	Yes	Yes

	abandoned.			
TransactionCommittedTotalCount	Returns the number of committed transactions.	Long	Yes	Yes
TransactionHeuristicsTotalCount	Returns the number of transactions that completed with a heuristic status.	Long	Yes	Yes
TransactionRolledBackAppTotalCount	Returns the number of transactions that were rolled back due to an application error.	Long	Yes	Yes
TransactionRolledBackResourceTotalCount	Returns the number of transactions that were rolled back due to a resource error.	Long	Yes	Yes
TransactionRolledBackSystemTotalCount	Returns the number of transactions that were rolled back due to an internal system error.	Long	Yes	Yes
TransactionRolledBackTimeoutTotalCount	Returns the number of transactions that were rolled back due to a timeout expiration.	Long	Yes	Yes
TransactionRolledBackTotalCount	Returns the number of transactions that were rolled back.	Long	Yes	Yes
TransactionTotalCount	Returns the total number of transactions processed. This total includes all committed, rolled back and heuristic transaction completions.	Long	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

JTA

The JTA MBean name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (TransactionTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

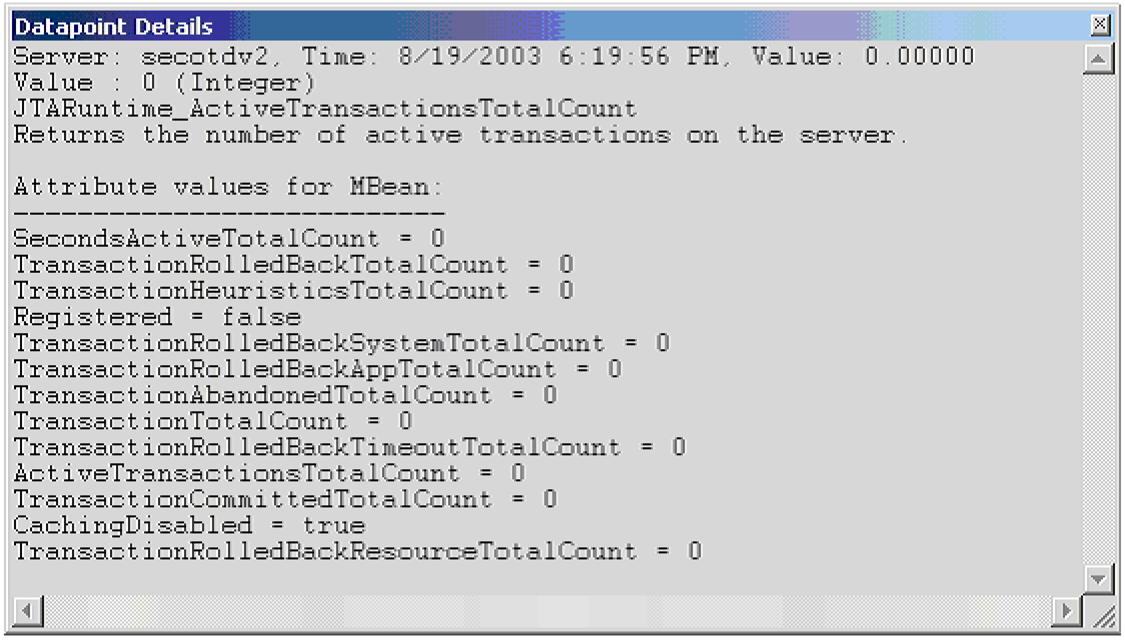
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:19:56 PM, Value: 0.00000
Value : 0 (Integer)
JTARuntime_ActiveTransactionsTotalCount
Returns the number of active transactions on the server.

Attribute values for MBean:
---------------------------
SecondsActiveTotalCount = 0
TransactionRolledBackTotalCount = 0
TransactionHeuristicsTotalCount = 0
Registered = false
TransactionRolledBackSystemTotalCount = 0
TransactionRolledBackAppTotalCount = 0
TransactionAbandonedTotalCount = 0
TransactionTotalCount = 0
TransactionRolledBackTimeoutTotalCount = 0
ActiveTransactionsTotalCount = 0
TransactionCommittedTotalCount = 0
CachingDisabled = true
TransactionRolledBackResourceTotalCount = 0
```

Interval

Recommended minimum is 5 minutes.

WebLogic JVM Runtime

The WebLogic JVM Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
HeapFreeCurrent	Returns the current amount of free memory (in bytes) in the JVM heap.	Long	Yes	Yes
HeapSizeCurrent	Returns the current size (in bytes) of the JVM heap.	Long	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

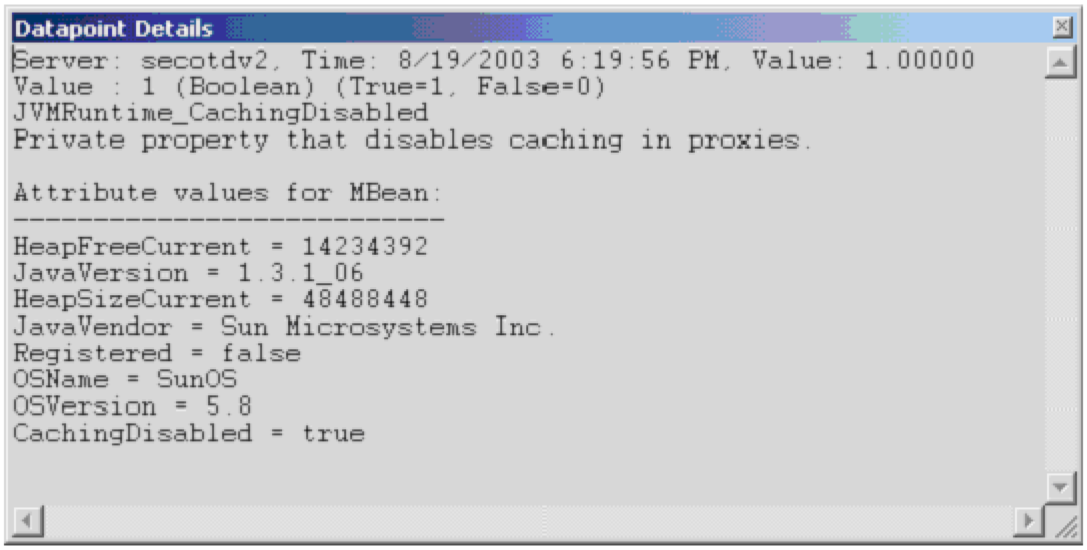
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:19:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
JVMRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
HeapFreeCurrent = 14234392
JavaVersion = 1.3.1_06
HeapSizeCurrent = 48488448
JavaVendor = Sun Microsystems Inc.
Registered = false
OSName = SunOS
OSVersion = 5.8
CachingDisabled = true
```

Interval

Recommended minimum is 5 minutes.

WebLogic Log Broadcaster Runtime

The WebLogic Log Broadcaster Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
MessagesLogged	Returns the total number of log messages generated by this instance of the WebLogic server.	Long	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Name

The name of the log broadcaster. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (MessagesLogged in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

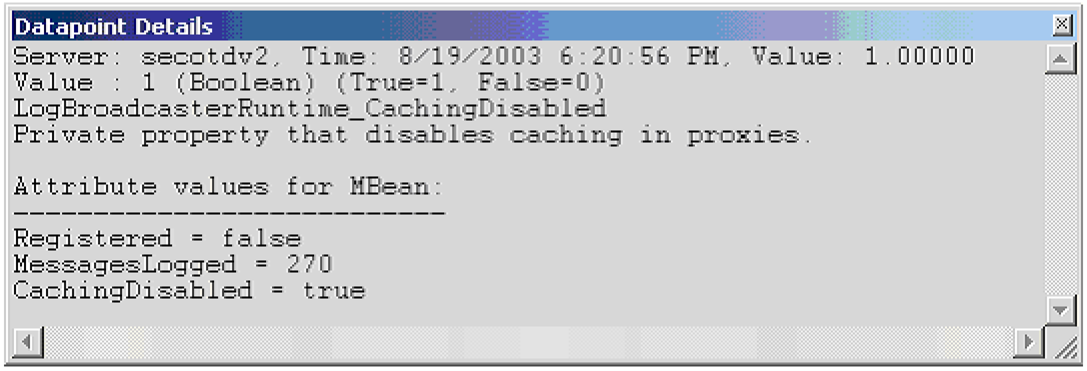
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:20:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
LogBroadcasterRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
Registered = false
MessagesLogged = 270
CachingDisabled = true
```

Interval

Recommended minimum is 5 minutes.

WebLogic Message Driven EJB Runtime

The WebLogic Message Driven EJB Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the

WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
JMSConnectionAlive	Returns the state of the EJB's JMS connection.	Boolean	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered	Boolean	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Application

The application prefix of the EJB ear. You can specify one or more application prefixes for monitoring. In any combination, select values from the discovered list, or enter values manually.

Component

The EJB component prefix of the EJB ear. You can specify one or more component prefixes for monitoring. In any combination, select values from the discovered list, or enter values manually.

Name

The remainder of the EJB name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

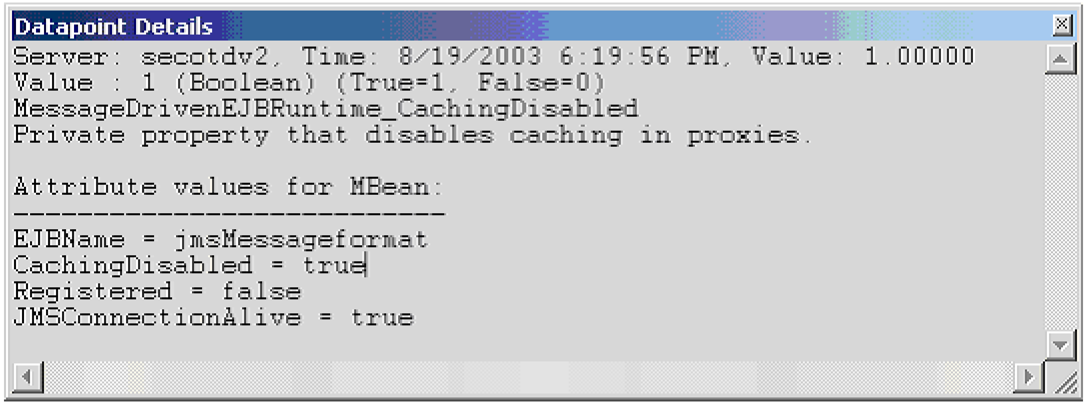
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:19:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
MessageDrivenEJBRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
EJBName = jmsMessageformat
CachingDisabled = true
Registered = false
JMSConnectionAlive = true
```

Interval

Recommended minimum is 5 minutes.

WebLogic Migratable Service Coordinator Runtime

The WebLogic Migratable Service Coordinator Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Service

The name of the migratable service coordinator. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

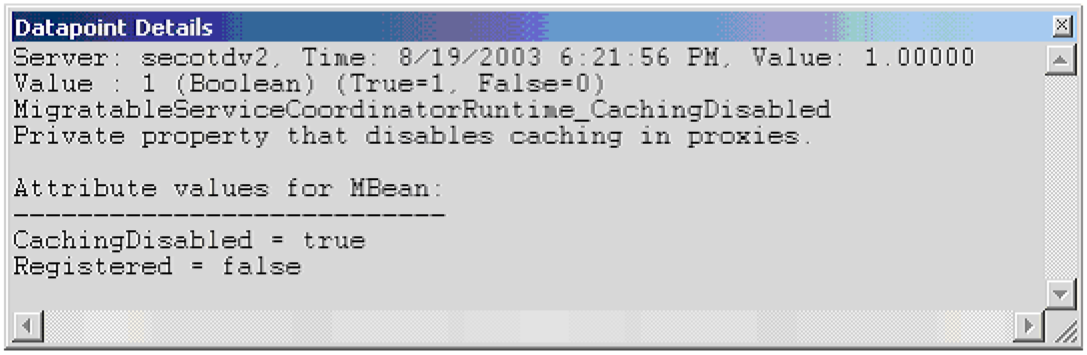
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:21:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
MigratableServiceCoordinatorRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
CachingDisabled = true
Registered = false
```

Interval

Recommended minimum is 5 minutes.

WebLogic Server Life Cycle Runtime

The WebLogic Server Life Cycle Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes
StateVal	Returns an integer that identifies the current state of the server. Values range from 0 to 8.	Integer	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Name

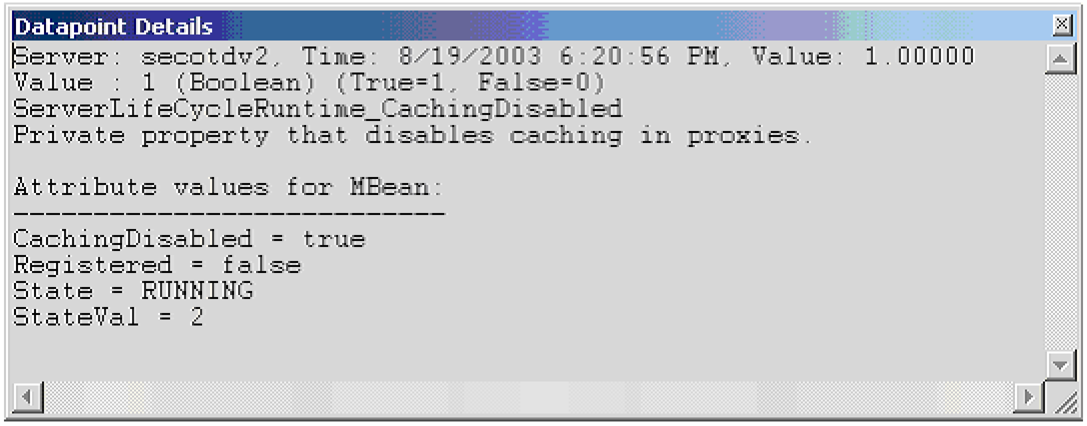
The application server name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:20:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
ServerLifeCycleRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
CachingDisabled = true
Registered = false
State = RUNNING
StateVal = 2
```

Interval

Recommended minimum is 5 minutes.

WebLogic Server Runtime

The WebLogic Server Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
ActivationTime	Returns the time when the server was started.	Long	Yes	Yes
AdministrationPort	Returns the administration port on which this server is listening for connections.	Integer	Yes	Yes
AdministrationPortEnabled	Returns whether the AdministrationPort is enabled on the server.	Boolean	Yes	Yes
AdminServer	Checks if the server is an administrator server.	Boolean	Yes	Yes
AdminServerListenPort	Returns the port on which admin server is listening for connections.	Integer	Yes	Yes
AdminServerListenPortSecure	Returns the secureType on which admin server is listening for connections.	Boolean	Yes	Yes
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
ListenPort	Returns the port on which this server is listening for connections.	Integer	Yes	Yes
ListenPortEnabled	Returns whether the default ListenPort is enabled on the server.	Boolean	Yes	Yes
OAMVersion	Returns the OAM version info. Indicates release level of this server.	Integer	Yes	Yes
OpenSocketsCurrentCount	Returns the current number sockets registered for socket muxing on this server.	Integer	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes
RestartsTotalCount	Returns the total number of restarts for this server since the cluster was last activated.	Integer	Yes	Yes
SocketsOpenedTotalCount	Returns the total number of registrations for socket muxing on this server.	Long	Yes	Yes
SSLListenPort	Returns the port on which this server is listening for SSL connections	Integer	Yes	Yes

SSLListenPortEnabled	Returns if the default lSSListenPort is enabled on the server.	Boolean	Yes	Yes
StateVal	Returns current state of the server.	Integer	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (SocketsOpenedTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

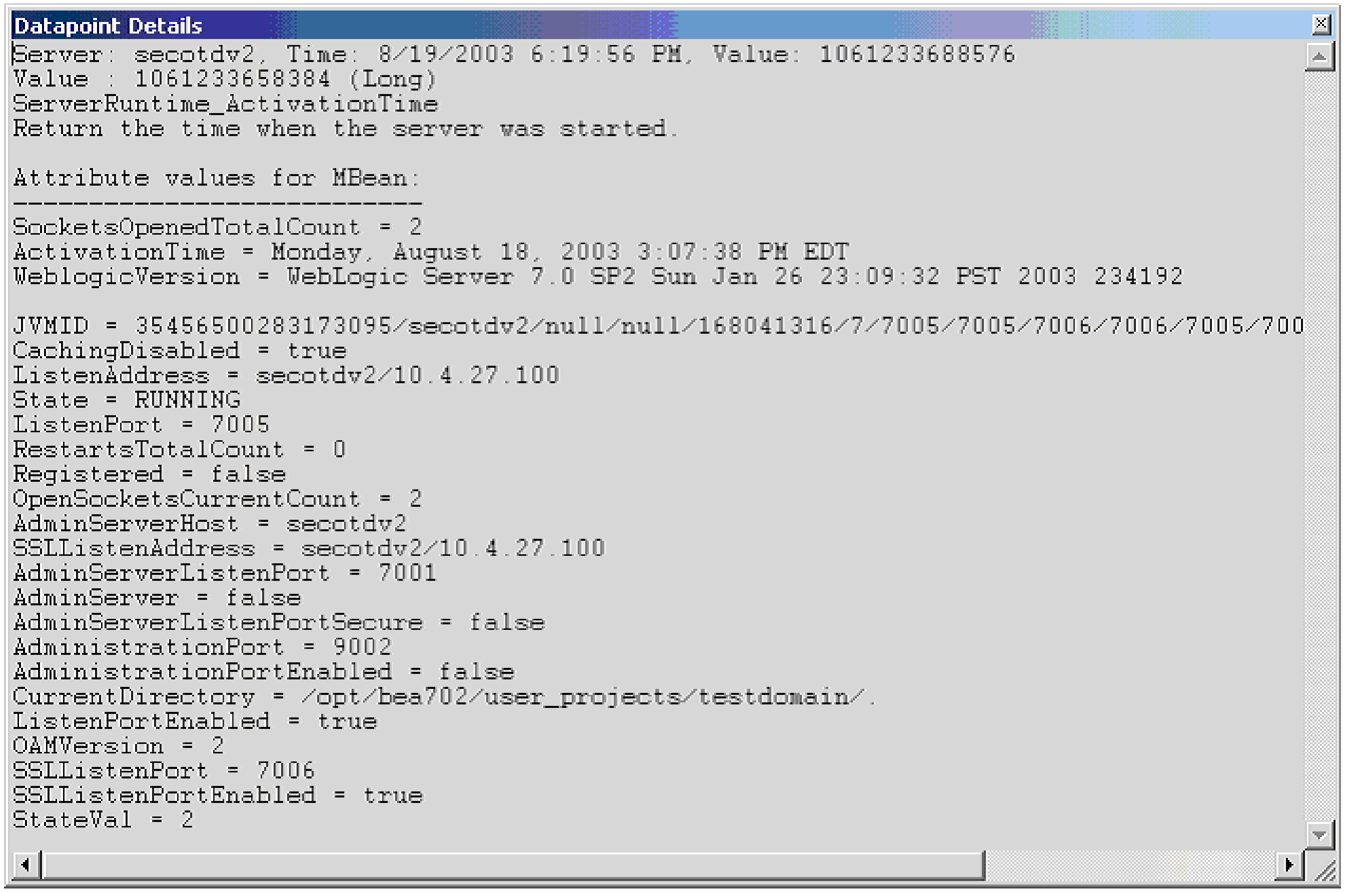
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



Datapoint Details

```
Server: secotdv2, Time: 8/19/2003 6:19:56 PM, Value: 1061233688576
Value : 1061233658384 (Long)
ServerRuntime_ActivationTime
Return the time when the server was started.

Attribute values for MBean:
---------------------------
SocketsOpenedTotalCount = 2
ActivationTime = Monday, August 18, 2003 3:07:38 PM EDT
WeblogicVersion = WebLogic Server 7.0 SP2 Sun Jan 26 23:09:32 PST 2003 234192

JVMID = 35456500283173095/secotdv2/null/null/168041316/7/7005/7005/7006/7006/7005/700
CachingDisabled = true
ListenAddress = secotdv2/10.4.27.100
State = RUNNING
ListenPort = 7005
RestartsTotalCount = 0
Registered = false
OpenSocketsCurrentCount = 2
AdminServerHost = secotdv2
SSLListenAddress = secotdv2/10.4.27.100
AdminServerListenPort = 7001
AdminServer = false
AdminServerListenPortSecure = false
AdministrationPort = 9002
AdministrationPortEnabled = false
CurrentDirectory = /opt/bea702/user_projects/testdomain/.
ListenPortEnabled = true
OAMVersion = 2
SSLListenPort = 7006
SSLListenPortEnabled = true
StateVal = 2
```

Interval

Recommended minimum is 5 minutes.

WebLogic Server Security Runtime

The WebLogic Server Security Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
InvalidLoginAttemptsTotalCount	Returns the cumulative number of invalid logins attempted on this server.	Long	Yes	Yes
InvalidLoginUsersHighCount	Returns the highest number of users with outstanding invalid login attempts for this server.	Long	Yes	Yes
LockedUsersCurrentCount	Returns the number of currently	Long	Yes	Yes

	locked users on this server.			
LoginAttemptsWhileLockedTotalCount	Returns the cumulative number of invalid logins attempted on this server while the user was locked.	Long	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes
UnlockedUsersTotalCount	Returns the number of times a user was unlocked on this server.	Long	Yes	Yes
UserLockoutTotalCount	Returns the cumulative number of user lockouts done on this server.	Long	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (InvalidLoginAttemptsTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

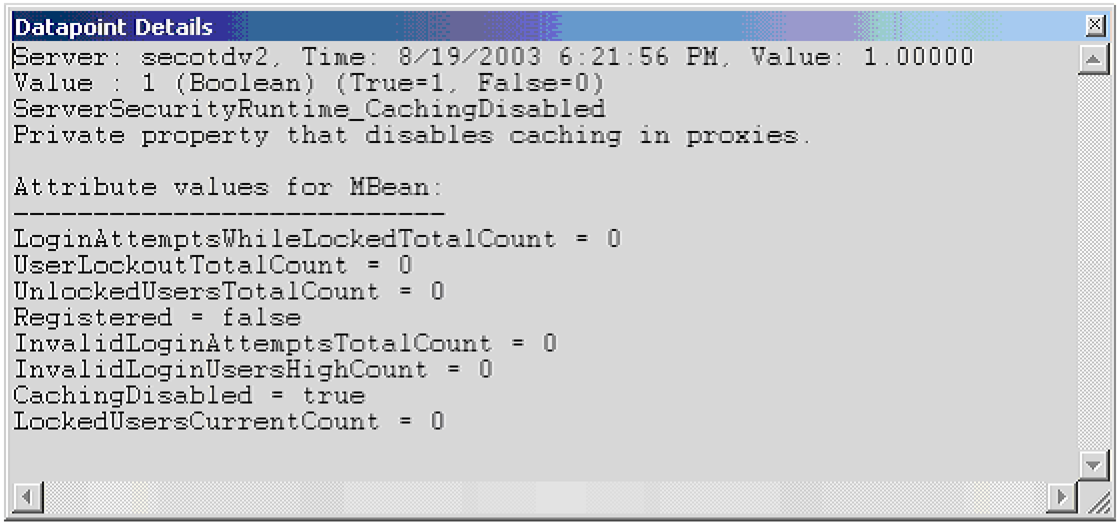
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:21:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
ServerSecurityRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
----------------------------
LoginAttemptsWhileLockedTotalCount = 0
UserLockoutTotalCount = 0
UnlockedUsersTotalCount = 0
Registered = false
InvalidLoginAttemptsTotalCount = 0
InvalidLoginUsersHighCount = 0
CachingDisabled = true
LockedUsersCurrentCount = 0
```

Interval

Recommended minimum is 5 minutes.

WebLogic Servlet Runtime

The WebLogic Application Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
ExecutionTimeAverage	Returns the average amount of time all invocations of the servlet have executed since it was created.	Integer	Yes	Yes
ExecutionTimeHigh	Returns the amount of time the single longest invocation of the servlet has executed since it was created.	Integer	Yes	Yes
ExecutionTimeLow	Returns the amount of time the single shortest invocation of the servlet has executed since it was created. Note: For the CounterMonitor, the difference option must be used.	Integer	Yes	Yes
ExecutionTimeTotal	Returns the amount of time all invocations of the servlet has executed since it was created.	Integer	Yes	Yes
InternalServlet	Returns whether this is an Internal Servlet.	Boolean	No	Yes
InvocationTotalCount	Returns the total number of times the	Integer	Yes	Yes

	servlet has been invoked.			
PoolMaxCapacity	Returns the maximum capacity of this servlet for single thread model servlets.	Integer	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered	Boolean	Yes	Yes
ReloadTotalCount	Returns the total number of times the servlet has been reloaded.	Integer	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Application

The application name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Servlet

The servlet name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (InvocationTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

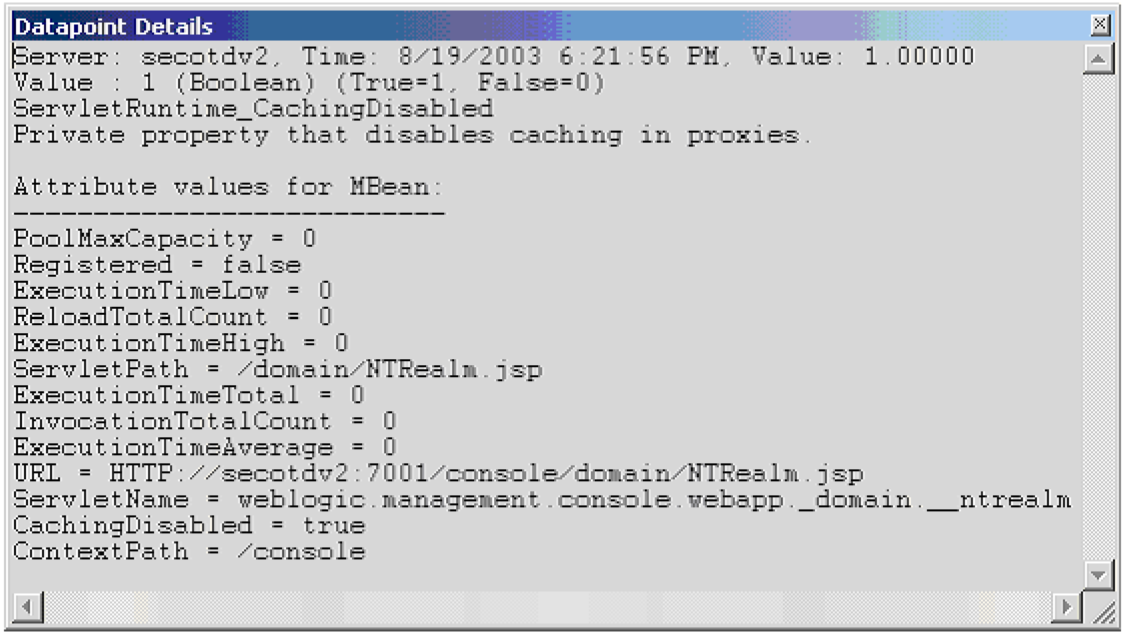
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.


Datapoint Details

```
Server: secotdv2, Time: 8/19/2003 6:21:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
ServletRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
PoolMaxCapacity = 0
Registered = false
ExecutionTimeLow = 0
ReloadTotalCount = 0
ExecutionTimeHigh = 0
ServletPath = /domain/NTRealm.jsp
ExecutionTimeTotal = 0
InvocationTotalCount = 0
ExecutionTimeAverage = 0
URL = HTTP://secotdv2:7001/console/domain/NTRealm.jsp
ServletName = weblogic.management.console.webapp._domain.__ntrealm
CachingDisabled = true
ContextPath = /console
```

Interval

Recommended minimum is 5 minutes.

WebLogic Stateful EJB Runtime

The WebLogic Stateful EJB Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	No
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	No

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Application

The application prefix of the EJB ear. You can specify one or more application prefixes for monitoring. In any combination, select values from the discovered list, or enter values manually.

Component

The EJB component prefix of the EJB ear. You can specify one or more component prefixes for monitoring. In any combination, select values from the discovered list, or enter values manually.

Name

The remainder of the EJB name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

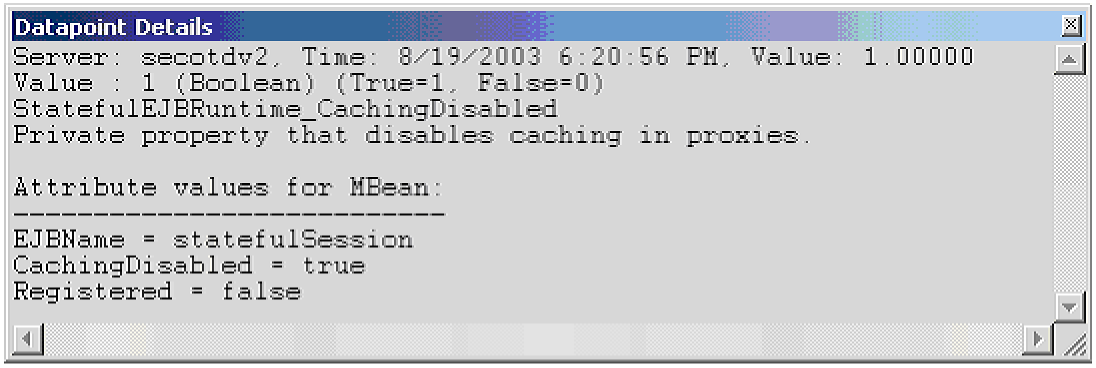
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:20:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
StatefulEJBRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
EJBName = statefulSession
CachingDisabled = true
Registered = false
```

Interval

Recommended minimum is 5 minutes.

WebLogic Stateless EJB Runtime

The WebLogic Stateless EJB Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX

Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Application

The application prefix of the EJB ear. You can specify one or more application prefixes for monitoring. In any combination, select values from the discovered list, or enter values manually.

Component

The EJB component prefix of the EJB ear. You can specify one or more component prefixes for monitoring. In any combination, select values from the discovered list, or enter values manually.

Name

The remainder of the EJB name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

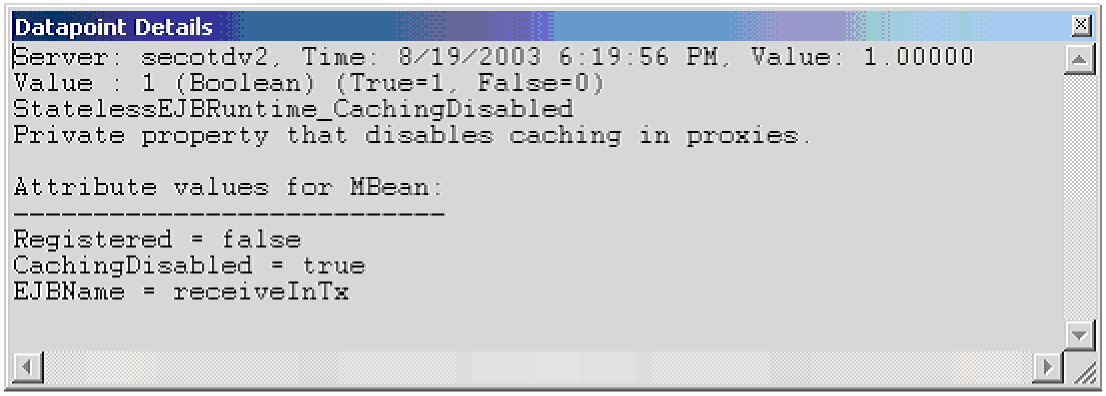
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:19:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
StatelessEJBRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
Registered = false
CachingDisabled = true
EJBName = receiveInTx
```

Interval

Recommended minimum is 5 minutes.

WebLogic Time Service Runtime

The WebLogic Time Service Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
ExceptionCount	Returns the total number of exceptions thrown while executing scheduled triggers.	Integer	Yes	Yes
ExecutionCount	Returns the total number of triggers executed.	Integer	Yes	Yes
ExecutionsPerMinute	Returns the average number of triggers executed per minute.	Integer	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes
ScheduledTriggerCount	Returns the number of currently active scheduled triggers.	Integer	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Name

The name of the time service. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (ExecutionCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

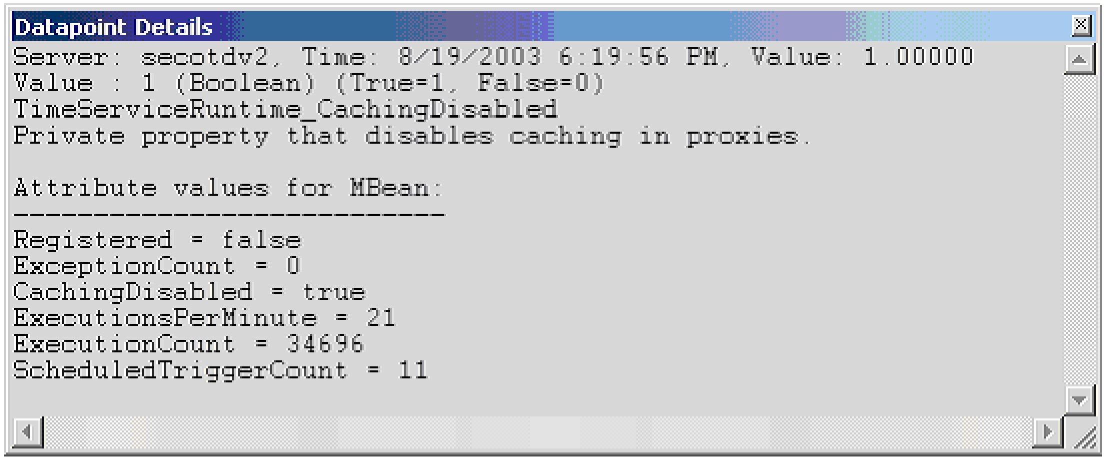
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:19:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
TimeServiceRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
Registered = false
ExceptionCount = 0
CachingDisabled = true
ExecutionsPerMinute = 21
ExecutionCount = 34696
ScheduledTriggerCount = 11
```

Interval

Recommended minimum is 5 minutes.

WebLogic Transaction Resource Runtime

The WebLogic Transaction Resource Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes
TransactionCommittedTotalCount	Returns the number of committed transactions.	Long	Yes	Yes
TransactionHeuristicCommitTotalCount	Returns the number of transactions for which this resource has returned a heuristic commit decision.	Long	Yes	Yes
TransactionHeuristicHazardTotalCount	Returns the number of transactions for which this resource has reported a heuristic hazard decision.	Long	Yes	Yes
TransactionHeuristicMixedTotalCount	Returns the number of transactions for which this resource has reported a heuristic mixed decision.	Long	Yes	Yes
TransactionHeuristicRollbackTotalCount	Returns the number of transactions for which this resource has returned a heuristic rollback decision.	Long	Yes	Yes
TransactionHeuristicsTotalCount	Returns the number of transactions that completed with a heuristic status.	Long	Yes	Yes
TransactionRolledBackTotalCount	Returns the number of transactions that were rolled back.	Long	Yes	Yes
TransactionTotalCount	Returns the total number of transactions processed. This total includes all committed, rolled back and heuristic transaction completions.	Long	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Transaction Runtime

The JTA runtime name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Component

The JTA component name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (TransactionCommittedTotalCount is one example in this counter category). Possible values are:

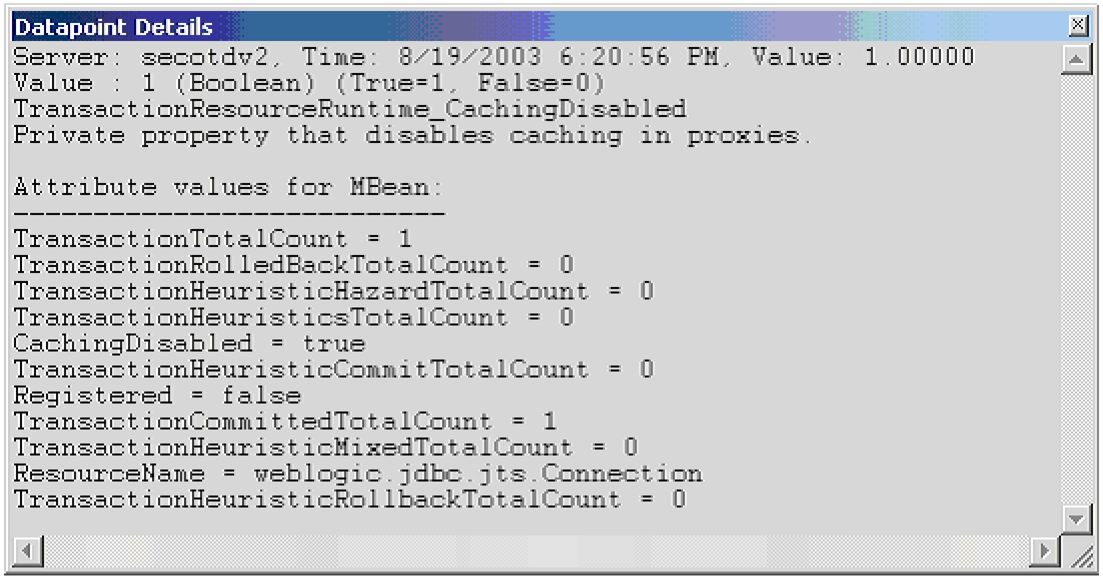
ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:20:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
TransactionResourceRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
TransactionTotalCount = 1
TransactionRolledBackTotalCount = 0
TransactionHeuristicHazardTotalCount = 0
TransactionHeuristicsTotalCount = 0
CachingDisabled = true
TransactionHeuristicCommitTotalCount = 0
Registered = false
TransactionCommittedTotalCount = 1
TransactionHeuristicMixedTotalCount = 0
ResourceName = weblogic.jdbc.jts.Connection
TransactionHeuristicRollbackTotalCount = 0
```

Interval

Recommended minimum is 5 minutes.

WebLogic Web App Component Runtime

The WebLogic Web App Component Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
DeploymentState	Returns the current deployment state of the module.	Integer	No	Yes
IndexDirectoryDisabled	Returns the directory indexing indicator configured in weblogic.xml.	Boolean	No	Yes
JSPDebug	Returns the JSP's debug/line numbers parameter values configured in weblogic.xml.	Boolean	No	Yes
JSPKeepGenerated	Returns the JSP's KeepGenerated parameter value configured in weblogic.xml.	Boolean	No	Yes
JSPPageCheckSecs	Returns the JSP's PageCheckSecs value configured in weblogic.xml.	Long	No	Yes
JSPVerbose	Returns the JSP's Verbose parameter value configured in weblogic.xml.	Boolean	No	Yes

OpenSessionsCurrentCount	Returns the current total number of open sessions in this component.	Integer	Yes	Yes
OpenSessionsHighCount	Returns the highest of the total number of open sessions in this server. The count starts at zero each time the server is activated. Note that this is an optimization method for a highly useful statistic that could be implemented less efficiently using change notification.	Integer	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes
ServletReloadCheckStatus	Returns the servlet reload check seconds configured in weblogic.xml.	Integer	No	Yes
SessionCookieMacAgentSecs	Returns the session's cookie max age configured for http sessions.	Integer	No	Yes
SessionInvalidationIntervalSecs	Returns the invalidation check timer interval configured for http sessions.	Integer	No	Yes
SessionMonitoringEnabled	Returns the session monitoring indicator configured in weblogic.xml.	Boolean	No	Yes
SessionsOpenedTotalCount	Returns the total number of sessions opened in this server.	Integer	Yes	Yes
SessionTimeoutSecs	Returns the timeout configured for http sessions.	Integer	No	Yes
SingleThreadServletPoolSize	Returns the single threaded servlet pool size configured in weblogic.xml.	Integer	No	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Application

The application name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

StatType

This parameter is available for counters that are returning a count or total (OpenSessionsCurrentCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

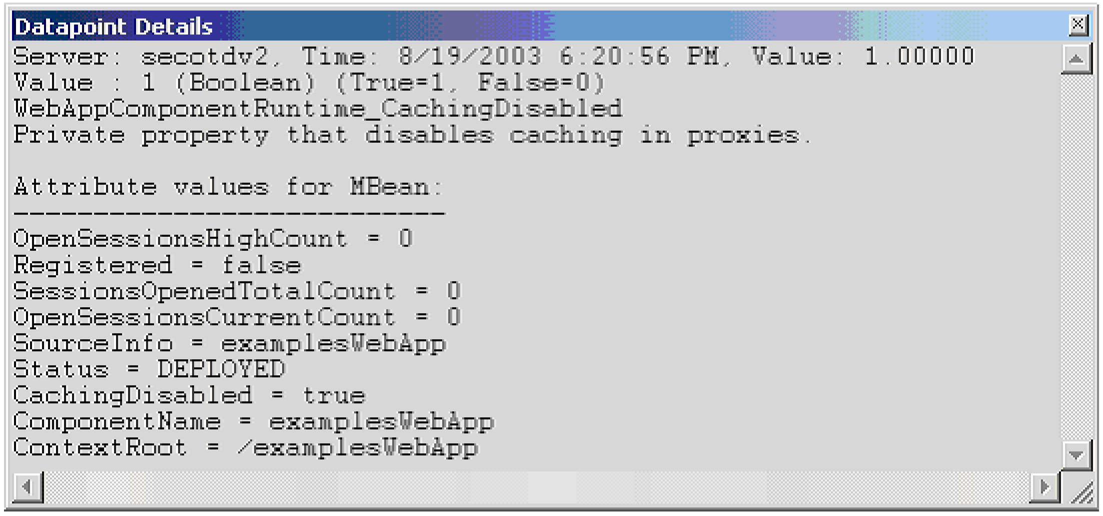
Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:20:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
WebAppComponentRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
----------------------------
OpenSessionsHighCount = 0
Registered = false
SessionsOpenedTotalCount = 0
OpenSessionsCurrentCount = 0
SourceInfo = examplesWebApp
Status = DEPLOYED
CachingDisabled = true
ComponentName = examplesWebApp
ContextRoot = /examplesWebApp
```

Interval

Recommended minimum is 5 minutes.

WebLogic Web Server Runtime

The WebLogic Web Server Runtime category includes the counters listed in the following table. Some of the counters listed in the table may not be available on your system. WebLogic counter categories, counter names, and parameters are dynamically discovered by processing the set of MBeans in the WebLogic JMX Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters	Description	Type	WL 7.x	WL 8.x

CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes
DefaultWebServer	Returns whether it is the defaultWebServer or a VirtualHost.	Boolean	No	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between the multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, or enter values manually.

Location

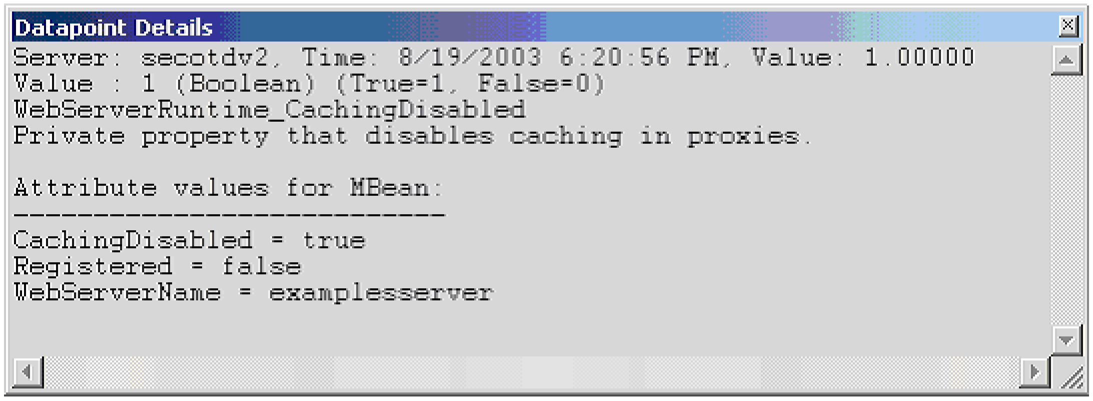
WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The PDP and IDP for a counter are displayed together in the following example. When parameters are defined by multiple values, a PDP and IDP are returned for each discovered combination of parameters.



```
Server: secotdv2, Time: 8/19/2003 6:20:56 PM, Value: 1.00000
Value : 1 (Boolean) (True=1, False=0)
WebServerRuntime_CachingDisabled
Private property that disables caching in proxies.

Attribute values for MBean:
---------------------------
CachingDisabled = true
Registered = false
WebServerName = examplesserver
```

Interval

Recommended minimum is 5 minutes.

WebLogic9 Counters

WebLogic9 Counters

ServerVantage provides the following dynamically discovered WebLogic9 remote counter categories. Each category provides counters that extend the monitoring of your WebLogic system. The categories, counters names, and parameters are all dynamically discovered by processing the set of MBeans available in the WebLogic JMX Server.

- WebLogic App Client Component Runtime
- WebLogic Application Runtime
- WebLogic Cluster Runtime
- WebLogic Component Runtime
- WebLogic Connector Component Runtime
- WebLogic Connector Connection Pool Runtime
- WebLogic Connector Connection Runtime
- WebLogic Connector Service Runtime
- WebLogic EJB Cache Runtime
- WebLogic EJB Component Runtime
- WebLogic EJB Locking Runtime
- WebLogic EJB Pool Runtime
- WebLogic EJB Timer Runtime
- WebLogic EJB Transaction Runtime
- WebLogic Entity Cache Cumulative Runtime
- WebLogic Entity Cache Current State Runtime
- WebLogic Execute Queue Runtime
- WebLogic Interception Component Runtime
- WebLogic JDBC Data Source Runtime
- WebLogic JDBC Data Source Task Runtime
- WebLogic JMS Component Runtime
- WebLogic JMS Connection Runtime
- WebLogic JMS Consumer Runtime
- WebLogic JMS Destination Runtime
- WebLogic JMS Durable Subscriber Runtime
- WebLogic JMS Pooled Connection Runtime
- WebLogic JMS Producer Runtime
- WebLogic JMS Remote Endpoint Runtime
- WebLogic JMS Runtime
- WebLogic JMS Server Runtime
- WebLogic JVM Runtime
- WebLogic Library Runtime
- WebLogic Log Broadcaster Runtime
- WebLogic Max Threads Constraint Runtime
- WebLogic Message Driven EJB Runtime
- WebLogic Min Threads Constraint Runtime
- WebLogic NonXA Resource Runtime
- WebLogic Persistent Store Connection Runtime
- WebLogic Persistent Store Runtime
- WebLogic Query Cache Runtime
- WebLogic Request Class Runtime
- WebLogic SAF Agent Runtime
- WebLogic SAF Remote Endpoint Runtime
- WebLogic Server Channel Runtime
- WebLogic Server Life Cycle Runtime
- WebLogic Server Life Cycle Task Runtime
- WebLogic Server Runtime
- WebLogic Server Security Runtime
- WebLogic Servlet Runtime
- WebLogic Task Runtime
- WebLogic Thread Pool Runtime
- WebLogic Transaction Name Runtime
- WebLogic Transaction Resource Runtime
- WebLogic User Lockout Manager Runtime
- WebLogic WAN Replication Runtime
- WebLogic Web App Component Runtime
- WebLogic Web Server Runtime
- WebLogic WLDF Archive Runtime
- WebLogic WLDF Data Access Runtime
- WebLogic WLDF Dbstore Archive Runtime

WebLogic_ApplicationRuntime

An application represents a J2EE Enterprise application packaged in an EAR file or EAR exploded directory. The EAR file or directory contains a set of components such as WAR, EJB, and RAR connector components, each of which can be deployed on one or more targets. A target is a server or a cluster.

ApplicationRuntime MBean encapsulates runtime information about a deployed Enterprise application.

The WebLogic_ApplicationRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
ActiveVersionState	An application can be the only version currently deployed, or it can have more than one version currently deployed, using the side-by-side deployment feature. If more than one version is deployed, only one version can be active. This attribute specifies the state in which the current application version is in. An application can be in an INACTIVE state, which means that it has not been activated yet or that there is more than one version of the application deployed (using side-by-side deployment) and this one is retiring. An application can be in ACTIVE_ADMIN state, which means that it is the currently active version for administrative channel requests. An application can be in ACTIVE state, which means that it is the currently active version for normal (non-administrative) channel requests. See weblogic.deploy.version.AppActiveVersionState for state values.	Integer

EAR	Returns true if the application deployment unit is an EAR file; returns false for WAR/JAR/RAR etc. deployments.	Boolean

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_AppClientComponentRuntime

This class represents a component runtime managed bean (MBean) for J2EE Application Client Containers.

The WebLogic_AppClientComponentRuntime category includes the counter listed in the following table. This counter may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of MBeans in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
DeploymentState	Returns the current deployment state of the module.	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

AppClientComponent

The name of the application client component. You can specify one or more application client components for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.

?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_ClusterRuntime

Use this class for monitoring a server's view of the members of a WebLogic cluster within a WebLogic domain.

The WebLogic_ClusterRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
AliveServerCount	Returns the current total number of alive servers in this cluster.	Integer
ForeignFragmentsDroppedCount	Returns the number of fragments that originated in foreign domains or clusters, using the same multi-cast address.	Long
FragmentsReceivedCount	Returns the total number of multi-cast messages	Long

	received on this server from the cluster.	
FragmentsSentCount	Returns the total number of multi-cast fragments sent from this server into the cluster.	Long
MulticastMessagesLostCount	Returns the total number of incoming multi-cast messages lost according to this server.	Long
PrimaryCount	Returns the number of objects that the local server hosts as primaries.	Long
ResendRequestsCount	Returns the number of state-delta messages resent because a receiving server in the cluster missed a message.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Cluster

The cluster name. You can specify one or more clusters for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (ResendRequestsCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_ComponentRuntime

This interface is the base class for all runtime MBeans that provide status of running modules.

The WebLogic_ComponentRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
DeploymentState	Returns the current deployment state of the module.	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_ConnectorComponentRuntime

This interface generates notifications about the deployment state of resource adapters. (Each resource adapter is represented by an instance of ConnectorComponentMBean .)

In two-phase deployment, if a resource adapter's state is PREPARED then it has achieved the first phase of deployment (everything is set up and all that remains is to enable a reference to the adapter). When the resource adapter is in an `ACTIVATED` state, it has achieved the second phase of deployment, in which applications can obtain a reference to the adapter.

A server instance creates an instance of this interface when it creates an instance of `weblogic.management.configuration.ConnectorComponentMBean`.

The WebLogic_ConnectorComponentRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
ActiveVersion	Returns true if this version is the active version. Returns true if this resource adapter is not versioned.	Boolean
AppDeploymentMBean	Returns the application deployment MBean for the connector component.	
ConnectionPoolCount	Returns the number of connection pools.	Integer
ConnectorComponentMBean	Returns the connector component MBean for the connector component.	Integer
DeploymentState	Returns the current deployment state of the module.	Integer
InboundConnectionsCount	Returns the number of inbound connections for the resource adapter.	Integer
SuspendedState	Returns resource adapter suspended state information. If getState() returns	Integer

	SUSPENDED, then getSuspendedState () returns an integer describing which functions of the resource adapter are suspended: One or more of INBOUND, OUTBOUND, or WORK (or ALL) or 0 for nothing suspended.	
Versioned	Returns true if the resource adapter is versioned.	Boolean

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

ConnectorComponent

The name of the connector component. You can specify one or more connector components for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (InboundConnectionsCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.

INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_ConnectorConnectionPoolRuntime

Use this class for monitoring a WebLogic connector connection pool.

The WebLogic_ConnectorConnectionPoolRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
ActiveConnectionsCurrentCount	Returns the current total active connections.	Integer
ActiveConnectionsHighCount	Returns the high-water mark of active connections in this connector	Integer

	pool since pool instantiation.	
AverageActiveUsage	Returns the running average usage of created connections active in the connector pool since the pool last shrunk.	Integer
CapacityIncrement	Returns the initial capacity configured for this connector connection pool.	Integer
CloseCount	Returns the number of connections closed for the connection pool.	Long
ConnectionIdleProfile Count	Returns the number of idle connection profiles stored for this pool. Deprecated.	Integer
ConnectionLeakProfileCount	Returns the number of leak connection profiles stored for this pool. Deprecated.	Integer
ConnectionProfilingEnabled	Indicates whether connection profiling is enabled for this pool. Deprecated.	Boolean
ConnectionsCreatedTotalCount	Returns the total number of connector connections created in this connector pool since pool instantiation.	Integer
ConnectionsDestroyedByErrorTotalCount	Returns the number of connections destroyed	Integer

	because of a received error event.	
ConnectionsDestroyedByShrinkingTotalCount	Returns the number of connections destroyed because of shrinking.	Integer
ConnectionsDestroyedTotalCount	Returns the total number of connector connections destroyed in this connector pool since pool instantiation.	Integer
ConnectionsMatchedTotalCount	Returns the total number of times a request for a connector connections was satisfied by using an existing created connection since pool instantiation.	Integer
ConnectionsRejectedTotalCount	Returns the total number of rejected requests for a connector connections in this connector pool since pool instantiation.	Integer
CurrentCapacity	Returns the pool size of this connector connection pool.	Long
FreeConnectionsCurrentCount	Returns the current total free connections.	Integer
FreeConnectionsHighCount	Returns the high-water mark of free connections in this connector pool since pool instantiation.	Integer

FreePoolSizeHighWaterMark	Returns the free pool size high-water mark of this connector connection pool.	Long
FreePoolSizeLowWaterMark	Returns the free pool size low-water mark of this connector connection pool.	Long
HighestNumWaiters	Returns the highest number of waiters.	Long
InitialCapacity	Returns the initial capacity configured for this connector connection pool.	Integer
LastShrinkTime	Returns the last time the pool was shrunk.	Long
LoggingEnabled	Indicates whether logging is enabled for this connector connection pool.	Boolean
MaxCapacity	Returns the maximum capacity configured for this connector connection pool.	Integer
MaxIdleTime	Returns the configured maximum idle time for this pool.	Integer
NumberDetectedIdle	Returns the total number of idle connections detected in the lifetime of this pool. Deprecated.	Integer
NumberDetectedLeaks	Returns the total number of leaked connections detected in the	Integer

	lifetime of this pool. Deprecated.	
NumUnavailableCurrentCount	Returns the number of unavailable connections.	Integer
NumUnavailableHighCount	Returns the highest number of connections unavailable at any given time.	Integer
NumWaiters	Returns the current number of waiters.	Long
NumWaitersCurrentCount	Returns the number of waiters.	Integer
PoolSizeHighWaterMark	Returns the pool size high-water mark of this connector connection pool.	Long
PoolSizeLowWaterMark	Returns the pool size low-water mark of this connector connection pool.	Long
ProxyOn	Returns a true flag if the proxy is on.	Boolean
RecycledTotal	Returns the total number of connector connections recycled in this connector connection pool since pool instantiation.	Integer
ShrinkCountDownTime	Returns the amount of time left (in minutes) until an attempt to shrink the pool is made.	Integer
ShrinkingEnabled	Specifies whether shrinking of this	Boolean

	connector connection pool is enabled.	
ShrinkPeriodMinutes	Returns the shrink period (in minutes) of this connector connection pool.	Integer
Testable	Indicates whether the connector connection pool is testable or not.	Boolean

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

ConnectorComponent

The name of the connector component. You can specify one or more connector components for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

ConnectorConnectionPool

The name connector connection pool. You can specify one or more connector connection pools for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (NumUnavailableHighCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_ConnectorConnectionRuntime

Use this class for monitoring individual WebLogic connector connections.

The WebLogic_ConnectorConnectionRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
ActiveHandlesCurrentCount	Returns the current total number of active connection handles for this connection.	Integer

ActiveHandlesHighCount	Returns the high-water mark of active connection handles for this connection since connection creation.	Integer
CreationDurationTime	Returns the time taken to create the connection.	Long
CurrentlyInUse	Indicates whether the connection is currently in use.	Boolean
Deletable	Indicates whether the connection can be closed manually through the console.	Boolean
HandlesCreatedTotalCount	Returns the total number of connection handles created for this connection since the connection creation.	Integer
Idle	Indicates whether the connection has been idle for a period extending beyond the configured maximum. Deprecated.	Boolean
InTransaction	Indicates whether the connection is currently in use in a transaction.	Boolean
LastUsage	Returns the last usage time stamp for the connection in milliseconds. Deprecated.	Long
ReserveDurationTime	Returns the time taken (in	Long

	seconds) to reserve this connection.	
ReserveTime	Returns the last time the connection was reserved.	Long
Shared	Indicates whether the connection is currently being shared by more than one invoker.	Boolean

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

ConnectorService

The name of the connector service. You can specify one or more connector services for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

ConnectionPools

The name of the connection pools. You can specify one or more connection pools for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

ConnectorConnection

The name of the connector connection. You can specify one or more connector connections for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (HandlesCreatedTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_ConnectorServiceRuntime

The counters contain runtime information that you can access at a connector service level, at a per resource adapter level, or at an overall level.

The WebLogic_ConnectorServiceRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
ActiveRACount	Returns the number of active resource adapters.	Integer
ConnectionPoolCurrentCount	Returns the number of connection pools	Integer

	in all active resource adapters. Deprecated.	
ConnectionPoolsTotalCount	Returns the total number of deployed and re-deployed connection pools instantiated since the server startup. Deprecated.	Integer
RACount	Returns the number of resource adapters deployed in the server. This count includes active and non-active resource adapters (in the case of versioned resource adapters being replaced by a new version).	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

ConnectorService

The name of the connector service. You can specify one or more connector services for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (ConnectionPoolsTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_EJBCacheRuntime

This interface contains accessor methods for all cache runtime information collected for an Enterprise JavaBean (EJB).

The WebLogic_EJBCacheRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
ActivationCount	Returns the total number of activated beans from this EJB	Long

	home.	
CacheAccessCount	Returns the total number of attempts to access a bean from the cache. The sum of the Cache Hit Count and Cache Miss Count may not add up to the cacheAccessCount in a running server because these metrics are retrieved using multiple calls and the counts could change between the calls.	Long
CachedBeansCurrentCount	Returns the total number of beans from this EJB home currently in the EJB cache.	Integer
CacheHitCount	Returns the total number of times an attempt to access a bean from the cache succeeded. The sum of the Cache Hit Count and Cache Miss Count may not add up to the CacheAccessCount in a running server because these metrics are retrieved using multiple calls and the counts could change between the calls. Deprecated. 28-Aug-2002. You may calculate the cache hit count by subtracting	Long

	the cache miss count from the cache access count.	
CacheMissCount	Returns the total number of times an attempt to access a bean from the cache failed. The sum of the Cache Hit Count and Cache Miss Count may not add up to the CacheAccessCount in a running server because these metrics are retrieved using multiple calls and the counts could change between the calls.	Long
PassivationCount	Returns the total number of passivated beans from this EJB home.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

EJBComponent

The name of the EJB component. You can specify one or more EJB components for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

EntityEJB

The name of the entity EJB. You can specify one or more entity EJBs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

EJBCache

The name of the EJB cache. You can specify one or more EJB caches for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (PassivationCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_EJBComponentRuntime

This interface is the top-level interface for all runtime information collected for an Enterprise JavaBean (EJB) module.

The WebLogic_EJBComponentRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
DeploymentState	Returns the current deployment state of the module.	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

EJBComponent

The name of the EJB component. You can specify one or more EJB components for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a

	wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- ! The primary data point (PDP) is the value returned for that counter.
- ! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_EJBLockingRuntime

This interface contains accessor methods for all lock manager runtime information collected for an Enterprise JavaBean (EJB).

The WebLogic_EJBLockingRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
LockEntriesCurrentCount	Returns the number of beans currently locked.	Integer
LockManagerAccessCount	Returns the total number of attempts to obtain a lock on a bean, including attempts to obtain a lock on a bean already locked on behalf of the client.	Long
TimeoutTotalCount	Returns the current number of threads that timed out	Long

	waiting for a lock on a bean.	
WaiterCurrentCount	Returns the current number of threads that waited for a lock on a bean.	Integer
WaiterTotalCount	Returns the total number of threads that waited for a lock on a bean.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

EJBComponent

The name of the EJB component. You can specify one or more EJB components for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatefulEJB

The name of the stateful EJB. You can specify one or more stateful EJBs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

EJBLocking

The name of the EJB lock. You can specify one or more EJB locks for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (WaiterTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_EJBPoolRuntime

This interface contains accessor methods for all free pool runtime information collected for an Enterprise JavaBean (EJB).

The WebLogic_EJBPoolRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
AccessTotalCount	Returns the total number of times an attempt was made to get an instance from the free pool.	Long

BeansInUseCount	Returns the total number of bean instances currently in use from the free pool. Deprecated. 28-Aug-2002. Use getBeansInUseCurrentCount() .	Integer
BeansInUseCurrentCount	Returns the number of bean instances currently in use from the free pool.	Integer
DestroyedTotalCount	Returns the total number of times a bean instance from this pool was destroyed because of a nonapplication exception being thrown from it.	Long
IdleBeansCount	Returns the total number of available bean instances in the free pool. Deprecated. 28-Aug-2002. Use getPooledBeansCurrentCount( ).	Integer
MissTotalCount	Returns the total number of times a failed attempt was made to get an instance from the free pool. An attempt to get a bean from the pool fails if there are no available instances in the pool.	Long
PooledBeansCurrentCount	Returns the current number of available bean instances in the free pool.	Integer
TimeoutTotalCount	Returns the total number of threads that timed out waiting for an available bean instance from the free pool.	Long
WaiterCurrentCount	Returns the number of threads currently waiting for an available bean instance from the free pool.	Integer
WaiterTotalCount	Returns the total number of threads currently waiting for an available bean instance from the free pool. Deprecated. 28-Aug-2002. Use getWaiterCurrentCount().	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

EJBComponent

The name of the EJB component. You can specify one or more EJB components for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatelessEJB

The name of the stateless EJB. You can specify one or more stateless EJBs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

EJBPool

The name of the EJB pool. You can specify one or more EJBs pools for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (WaiterTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.

^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_EJBTimerRuntime

This interface contains accessor methods for all Enterprise JavaBean (EJB) timer runtime information collected for an EJB.

The WebLogic_EJBTimerRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
ActiveTimeCount	Returns the current number of active timers for this EJB.	Integer
CancelledTimerCount	Returns the total number of timers that explicitly cancelled for this EJB.	Long
DisabledTimerCount	Returns the current number of timers temporarily disabled for this EJB.	Integer
TimeoutCount	Returns the total number of successful time-	Long

	out notifications made for this EJB.	

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

EJBComponent

The name of the EJB component. You can specify one or more EJB components for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatelessEJB

The name of the stateless EJB. You can specify one or more stateless EJBs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

EJBTimer

The name of the EJB timer. You can specify one or more EJB timers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (TimeoutCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_EJBTransactionRuntime

This interface contains accessor methods for all transaction runtime information collected for an Enterprise JavaBean (EJB).

The WebLogic_EJBTransactionRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
TransactionsCommittedTotalCount	Returns the total number of transactions committed for this EJB.	Long
TransactionsRolledBackTotalCount	Returns the total number of transactions rolled back for this EJB.	Long
TransactionsTimedOutTotalCount	Returns the total number of timed-out transactions	Long

	for this EJB.	

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

EJBComponent

The name of the EJB component. You can specify one or more EJB components for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

EntityEJB

The name of the entity EJB. You can specify one or more entity EJBs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

EJBTransaction

The name of the EJB transaction. You can specify one or more EJB transactions for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (TransactionsTimedOutTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_EntityCacheCumulativeRuntime

Use this class for monitoring an EXtensible Markup Language (XML) cache.

The WebLogic_EntityCacheCumulativeRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
AvgEntrySizeDiskPurged	Returns the cumulative average size of entries purged from the disk cache.	Double
AvgEntrySizeMemoryPurged	Returns the average size of all entries purged from the memory.	Double
AvgPercentPersistent	Returns the current average percentage of entries in the	Double

	entity cache persisted to the disk cache.	
AvgPercentTransient	Returns the current average percentage of entries in the entity cache that are transient or have not been persisted.	Double
AvgPer EntryDiskSize	Returns the current average size of the entries in the entity disk cache.	Double
AvgPerEntryMemorySize	Returns the current average size of the entries in the entity memory cache.	Double
AvgTimeout	Returns the average amount of time that the entity cache has timed-out when trying to retrieve an entity.	Double
DiskPurgesPerHour	Returns the cumulative average number of purges from the disk cache per hour.	Double
MaxEntryMemorySize	Returns the current maximum size of the entries in the entity memory cache.	Long
MaxEntryTimeout	Returns the largest time-out value for any current entry in the entity cache.	Double
MemoryPurgesPerHour	Returns the cumulative average number of entries purged	Double

	from the entity cache.	
MinEntryMemorySize	Returns the current minimum size of the entries in the entity memory cache.	Long
MinEntryTimeout	Returns the smallest time-out value for any current entry in the entity cache.	Double
PercentRejected	Returns the cumulative percent of the potential entries rejected to the entity cache.	Double
TotalCurrentEntries	Returns the total current number of entries in the entity cache.	Long
TotalItemsDiskPurged	Returns the total number of items purged from the disk cache.	Long
TotalItemsMemoryPurged	Returns the cumulative number of items purged from the entity cache.	Long
TotalNumberDiskPurges	Returns the total number of entries purged from the disk cache.	Long
TotalNumberMemoryPurges	Returns the cumulative total number of entries purged from the entity cache.	Long
TotalNumberOfRejections	Returns the cumulative total number of entries rejected from the entity cache for the current session.	Long

TotalNumberOfRenewals	Returns the cumulative total number of entries refreshed in the entity cache.	Long
TotalPersistentCurrentEntries	Returns the total current number of entries in the cache having been persisted to disk.	Long
TotalSizeOfRejections	Returns the cumulative total size of the rejections from the entity cache.	Long
TotalTransientCurrentEntries	Returns the total current number of transient (not yet persisted to disk) entries in the entity cache.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

EntityCacheCumulative

The name of the entity cache cumulative. You can specify one or more entity cache cumulatives for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_EntityCacheCurrentStateRuntime

Use this class for monitoring the size and usage of an EXtensible Markup Language (XML) cache.

The WebLogic_EntityCacheCurrentStateRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
AvgPercent Persistent	Returns the current average percentage of entries in the entity cache persisted to the disk cache.	Double
AvgPercentTransient	Returns the current average percentage of entries in the entity cache that are transient (not	Double

	yet persisted).	
AvgPer EntryDiskSize	Returns the current average size of the entries in the entity disk cache.	Double
AvgPerEntryMemorySize	Returns the current average size of the entries in the entity memory cache.	Double
AvgTimeout	Returns the average amount of time that the entity cache has timed-out when trying to retrieve an entity.	Double
DiskUsage	Returns the current size of the entity disk cache.	Long
MaxEntryMemorySize	Returns the current maximum size of the entries in the entity memory cache.	Long
MaxEntryTimeout	Returns the largest time-out value for any current entry in the entity cache.	Double
MemoryUsage	Returns the current size of the entity memory cache.	Long
MinEntryMemorySize	Returns the current minimum size of the entries in the entity memory cache.	Long
MinEntryTimeout	Returns the smallest time-out value for any current entry in	Double

	the entity cache.	
TotalCurrentEntries	Returns the total current number of entries in the entity cache.	Long
TotalPersistentCurrentEntries	Returns the total current number of entries in the cache persisted to disk.	Long
TotalTransientCurrentEntries	Returns the total current number of transient entries in the entity cache.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

EntityCacheCurrentState

The name of the entity cache current state. You can specify one or more entity cache current states for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.

?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_ExecuteQueueRuntime

Use this bean to monitor an execute queue and its associated thread pool.

The WebLogic_ExecuteQueueRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
ExecuteThreadCurrentIdleCount	Returns the number of idle threads assigned to the queue.	Integer
ExecuteThreadTotalCount	Returns the total number of execute threads assigned to the queue.	Integer
PendingRequestCurrentCount	Returns the number of waiting requests in the queue.	Integer
PendingRequestOldestTime	Returns the time that the longest waiting request	Long

	was placed in the queue.	
ServicedRequestTotalCount	Returns the number of requests processed by the queue.	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

ExecuteQueue

The name of the execute queue. You can specify one or more execute queues for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (ServicedRequestTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to

represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_InterceptionComponentRuntime

This class is the top-level interface for all runtime information collected for an Interception module.

The WebLogic_InterceptionComponentRuntime category includes the counter listed in the following table. This counter may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
DeploymentState	Returns the current deployment state of the module.	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

InterceptionComponent

The name of the interception component. You can specify one or more interception components for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JDBCDataSourceRuntime

Use this class for monitoring a WebLogic Java DataBase Connectivity (JDBC) data source and its associated connection pool.

The WebLogic_JDBCDataSourceRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
ActiveConnectionsAverageCount	Returns the	Integer

	average number of active connections in this instance of the data source. Active connections are connections in use by an application.	
ActiveConnectionsCurrentCount	Returns the number of connections currently in use by applications.	Integer
ActiveConnectionsHighCount	Returns the highest number of active database connections in this instance of the data source since data source instantiation. Active connections are connections in use by an application.	Integer
ConnectionDelayTime	Returns the average time (in milliseconds) needed to create a physical connection to the database. The value is calculated as a summary of all times to connect divided by the total number of connections.	Integer
ConnectionsTotalCount	Returns the cumulative total number of database connections created in this data source since data source deployment.	Integer
CurrCapacity	Returns the current count of JDBC connections in the connection	Integer

	pool in the data source.	
CurrCapacityHighCount	Returns the highest number of database connections available or in use (current capacity) in this instance of the data source since data source deployment.	Integer
DeploymentState	Returns the current deployment state of the module.	Integer
Enabled	Indicates whether the data source is enabled or disabled: Returns true if the data source is enabled; returns false if the data source is disabled.	Boolean
FailedReserveRequestCount	Returns the cumulative, running count of requests for a connection from this data source that could not be fulfilled.	Long
FailuresToReconnectCount	Returns the number of times that the data source attempted to refresh a database connection and failed. Failures may occur when the database is unavailable or when the network connection to the database is interrupted.	Integer
HighestNumAvailable	Returns the highest number of	Integer

	database connections available at any time in this instance of the data source since data source deployment.	
HighestNumUnavailable	Returns the highest number of database connections unavailable (in use or being tested by the system) in this instance of the data source since data source deployment.	Integer
LeakedConnectionCount	Returns the number of leaked connections. A leaked connection is a connection reserved from the data source but not returned to the data source by calling close().	Integer
NumAvailable	Returns the number of database connections currently available (not in use) in this data source.	Integer
NumUnavailable	Returns the number of database connections currently unavailable (in use or being tested by the system) in this instance of the data source.	Integer
PrepStmtCacheAccessCount	Returns the cumulative, running count of the number of times that the	Long

	statement cache was accessed.	
PrepStmtCacheAddCount	Returns the cumulative, running count of the number of statements added to the statement cache. Each connection in the connection pool has its own cache of statements. This number is the sum of the number of statements added to the caches for all connections in the connection pool.	Long
PrepStmtCacheCurrentSize	Returns the number of prepared and callable statements currently cached in the statement cache. Each connection in the connection pool has its own cache of statements. This number is the sum of the number of statements in the caches for all connections in the connection pool.	Integer
PrepStmtCacheDeleteCount	Returns the cumulative, running count of statements discarded from the cache. Each connection in the connection pool has its own cache of statements. This number is the sum of the number of statements that were discarded from the caches for all connections in	Long

	the connection pool.	
PrepStmtCacheHitCount	Returns the cumulative, running count of the number of times that statements from the cache were used.	Integer
PrepStmtCacheMissCount	Returns the number of times that a statement request could not be satisfied with a statement from the cache.	Integer
ReserveRequestCount	Returns the cumulative, running count of requests for a connection from this data source.	Long
WaitingForConnectionCurrentCount	Returns the number of connection requests waiting for a database connection.	Integer
WaitingForConnectionFailureTotal	Returns the cumulative, running count of requests for a connection from this data source that waited before getting a connection and eventually failed to get a connection.	Long
WaitingForConnectionHighCount	Returns the highest number of application requests concurrently waiting for a connection from this instance of the data source.	Integer

WaitingForConnectionSuccessTotal	Returns the cumulative, running count of requests for a connection from this data source that had to wait before getting a connection and eventually succeeded in getting a connection.	Long
WaitingForConnectionTotal	Returns the cumulative, running count of requests for a connection from this data source that had to wait before getting a connection, including those that eventually got a connection and those that did not get a connection.	Long
WaitSecondsHighCount	Returns the longest connection reserve wait time in seconds.	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JDBCDataSource

The name of the JDBC data source. You can specify one or more JDBC data sources for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (ReserveRequestCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JDBCDataSourceTaskRuntime

Information for this WebLogic_JDBCDataSourceTaskRuntime (Java Data Base Connectivity) category is available only at runtime. To view information about this MBean, start the host WebLogic server instance,and then use WebLogic Scripting Tool in interactive mode. Or, you may use some other category browser.

The WebLogic_JDBCDataSourceTaskRuntime category includes the counters listed in the following list. Some of these counters may not be available on your system. ServerVantage dynamically discovers

WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
BeginTime	Returns the task start time.	Long
EndTime	Returns the task completion time. A value of -1 indicates that the task is currently running.	Long
Running	Indicates whether the task is still running.	Boolean
SystemTask	Indicates whether this task was initiated by the server versus a user.	Boolean

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JDBCService

The name of the JDBC service. You can specify one or more JDBC services for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JDBCDataSource

The name of the JDBC data source. You can specify one or more JDBC data sources for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JDBCDataSourceTask

The name of the JDBC data source task. You can specify one or more JDBC data source tasks for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JMSComponentRuntime

This class is the top-level interface for all runtime information collected for a Java Messaging Service (JMS) module.

The WebLogic_JMSComponentRuntime category includes the counter listed in the following table. This counter may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
DeploymentState	Returns the current deployment state	Integer

	of the module.	

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSComponent

The name of the JMS component. You can specify one or more JMS components for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JMSConnectionRuntime

Use this class for monitoring a WebLogic Java Messaging Service (JMS) connection.

The WebLogic_JMSConnectionRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
SessionsCurrentCount	Returns the current number of sessions for this connection.	Long
SessionsHighCount	Returns the peak number of sessions for this connection since the last reset.	Long
SessionsTotalCount	Returns the total number of sessions for this connection since the last reset.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMS

The JMS name. You can specify one or more JMSs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSConnection

The name of the JMS connection. You can specify one or more JMS connections for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (SessionsTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JMSConsumerRuntime

Use this class for monitoring a WebLogic Java Messaging Service (JMS) consumer.

The WebLogic_JMSConsumerRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the

WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
Active	Indicates whether the consumer is active. A consumer is active if it has a message listener set up or a synchronous receive in progress.	Boolean
BytesPendingCount	Returns the number of bytes pending (uncommitted and unacknowledged) by this consumer.	Long
BytesReceivedCount	Returns the number of bytes received by this consumer since the last reset.	Long
Durable	Indicates whether the consumer is durable.	Boolean
MessagesPendingCount	Returns the number of messages pending (uncommitted and unacknowledged) by this consumer.	Long
MessagesReceivedCount	Returns the number of messages received by this consumer since the last reset.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter

dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMS

The Java Messaging Service (JMS) name. You can specify one or more JMSs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSConnection

The name of the JMS connection. You can specify one or more JMS connections for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSSession

The name of the JMS session. You can specify one or more JMS sessions for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSConsumer

The name of the JMS consumer. You can specify one or more JMS consumers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (MessagesReceivedCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to

represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JMSDestinationRuntime

Use this class for monitoring a WebLogic Java Messaging Service (JMS) destination (topic or queue).

The WebLogic_JMSDestinationRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
BytesCurrentCount	Returns the current number of bytes stored (not pending bytes) in the destination.	Long
BytesHighCount	Returns the peak number of bytes stored in the destination since the last reset.	Long
BytesPendingCount	Returns the number of pending bytes stored in the destination. Pending bytes are over and above the current number of bytes.	Long
BytesReceivedCount	Returns the number of bytes received in this destination since the last reset.	Long
BytesThresholdTime	Returns the amount of time in the threshold condition since the last reset.	Long
ConsumersCurrentCount	Returns the current number of consumers accessing this destination.	Long
ConsumersHighCount	Returns the peak number of consumers accessing this destination since the last reset.	Long
ConsumersTotalCount	Returns the total number of consumers accessing this destination since the last reset.	Long

ConsumptionPaused	Indicates the consumption pause state of the destination.	Boolean
InsertionPaused	Returns the insertion pause state of the destination.	Boolean
MessagesCurrentCount	Returns the current number of messages (not pending messages) in the destination.	Long
MessagesHighCount	Returns the peak number of messages in the destination since the last reset.	Long
MessagesMovedCurrentCount	Returns the number of messages moved from the destination.	Long
MessagesPendingCount	Returns the number of pending messages in the destination. Pending messages are over and above the current number of messages. A pending message is one sent in a transaction and not committed or one forwarded but not acknowledged.	Long
MessagesReceivedCount	Returns the number of messages received in this destination since the last reset.	Long
MessagesThresholdTime	The amount of time (in seconds) in the threshold condition since the last reset.	Long
Paused	Indicates whether the destination is paused at the current time. Deprecated. 9.0.0.0. Replaced by JMSDestinationRuntimeMBean#isProductionPaused.	Boolean

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSServer

The name of the JMS server. You can specify one or more JMS servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSDestination

The name of the JMS destination You can specify one or more JMS destinations for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (MessagesReceivedCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JMSDurableSubscriberRuntime

Use this class for monitoring a WebLogic Java Messaging Service (JMS) durable subscriber.

The WebLogic_JMSDurableSubscriberRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type

Active	Indicates whether this subscription is being used by a durable subscriber.	Boolean
BytesCurrentCount	Returns the number of bytes received by this durable subscriber.	Long
BytesPendingCount	Returns the number of bytes pending by this durable subscriber.	Long
MessagesCurrentCount	Returns the number of messages still available by this durable subscriber.	Long
MessagesDeletedCurrentCount	Returns the number of messages deleted from the destination.	Long
MessagesHighCount	Returns the peak number of messages for this durable subscriber since the last reset.	Long
MessagesMovedCurrentCount	Returns the number of messages moved from the destination.	Long
MessagesPendingCount	Returns the number of messages pending (uncommitted and unacknowledged) by this durable subscriber.	Long
MessagesReceivedCount	Returns the number of messages received	Long

	by this durable subscriber since the last reset.	
NoLocal	Indicates the value of the noLocal Boolean for this durable subscriber.	Boolean

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMS

The JMS name. You can specify one or more JMSs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSServers

The name of the JMS server. You can specify one or more JMS servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSDestinations

The JMS destination name. You can specify one or more JMS destinations for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSDurableSubscribers

The name of the JMS durable subscriber. You can specify one or more JMS durable subscribers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (BytesCurrentCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.

INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JMSPooledConnectionRuntime

Use this class for monitoring pooled Java Messaging Service (JMS) connections. A pooled JMS connection is a session pool used by Enterprise JavaBeans (EJBs) and servlets that use a resource-reference element in their EJB or Servlet deployment descriptor to define their JMS connection factories.

The WebLogic_JMSPooledConnectionRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
AverageReserved	Returns the average number of JMS sessions in use in this instance of the session pool since its instantiation, which generally	Integer

	occurs when an EJB or servlet is deployed that requires the session pool.	
CreationDelayTime	Returns the average amount of time needed to create each JMS session in the session pool.	Integer
CurrCapacity	Returns the current capacity of the session pool, which is always less than or equal to the maximum capacity of JMS sessions.	Integer
HighestNumAvailable	Returns the peak number of available JMS sessions in this instance of the session pool since its instantiation.	Integer
HighestNumReserved	Returns the peak number of concurrent JMS sessions reserved for this instance of the session pool since its instantiation.	Integer
HighestNumUnavailable	Returns the peak number of unusable JMS sessions in this instance of the session pool since its instantiation.	Integer
HighestNumWaiters	Returns the peak number of threads waiting to retrieve a JMS session in this instance of the session pool since its instantiation.	Integer

HighestWaitSeconds	Returns the peak number of seconds that an application waited to retrieve a JMS session in this instance of the session pool since its instantiation.	Integer
MaxCapacity	Returns the maximum number of JMS sessions that may be allocated using the session pool.	Integer
NumAvailable	Returns the number of available JMS sessions in the session pool not currently in use.	Integer
NumConnectionObjects	Returns the number of JMS connections that back this session pool. This value may be greater than one if different sessions were created using different combinations of a user name and password to contact the JMS server.	Integer
NumFailuresToRefresh	Returns the number of failed attempts to create a JMS session in the session pool.	Integer
NumLeaked	Returns the number of JMS sessions removed from the session pool but not returned.	Integer

NumReserved	Returns the number of JMS sessions currently in use.	Integer
NumUnavailable	Returns the number of JMS sessions in the session pool not currently usable for an unknown reason.	Integer
NumWaiters	Returns the number of threads waiting to retrieve a JMS session from the session pool.	Integer
TotalNumAllocated	Returns the total number of JMS sessions allocated by this session pool in this instance of the session pool since its instantiation.	Integer
TotalNumDestroyed	Returns the total number of JMS sessions created and then destroyed in this instance of the session pool since its instantiation.	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JMSProducerRuntime

Use this class for monitoring a WebLogic Java Messaging Service (JMS) producer.

The WebLogic_JMSProducerRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
BytesPendingCount	Returns the number of bytes pending (uncommitted and unacknowledged) by this producer.	Long
BytesSentCount	Returns the number of bytes	Long

	sent by this producer since the last reset.	
MessagesPendingCount	Returns the number of messages pending (uncommitted and unacknowledged) by this producer.	Long
MessagesSentCount	Returns the number of messages sent by this producer since the last reset.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMS

The JMS name. You can specify one or more JMSs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSServers

The name of the JMS server. You can specify one or more JMS server for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSSessionPool

The name of the JMS session pool. You can specify one or more JMS session pools for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (MessagesSentCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JMSRemoteEndpointRuntime

Use this class for monitoring a WebLogic Store-and-Forward (SAF) remote endpoint for a Java Messaging Service (JMS) imported destination.

The WebLogic_JMSRemoteEndpointRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
BytesCurrentCount	Returns the current number of bytes, excluding	Long

	pending bytes.	
BytesHighCount	Returns the peak number of bytes since the last reset.	Long
BytesPendingCount	Returns the number of pending bytes. Pending bytes are over and above the current number of bytes.	Long
BytesReceivedCount	Returns the number of bytes received since the last reset.	Long
BytesThresholdTime	Returns the amount of time (in seconds) in the threshold condition since the last reset.	Long
DowntimeHigh	Returns the longest time (in seconds) that the remote endpoint has not been available since the last reset.	Long
DowntimeTotal	Returns the total time (in seconds) that the remote endpoint has not been available since the last reset.	Long
FailedMessagesTotal	Returns the total number of messages that have failed to be forwarded since the last reset.	Long
MessagesCurrentCount	Returns the current number of messages, including pending	Long

	messages.	
MessagesHighCount	Returns the peak number of messages since the last reset.	Long
MessagesPendingCount	Returns the number of pending messages. Pending messages are over and above the current number of messages. A pending message is one sent in a transaction and not committed or one forwarded but not acknowledged.	Long
MessagesReceivedCount	Returns the number of messages received since the last reset.	Long
MessagesThresholdTime	Returns the amount of time (in seconds) in the threshold condition since the last reset.	Long
PausedForForwarding	Indicates whether the remote endpoint is not forwarding messages at the current time.	Boolean
PausedForIncoming	Indicates whether a remote endpoint is not accepting new messages at the current time.	Boolean
UptimeHigh	Returns the longest time (in seconds) that the remote endpoint has been	Long

	available since the last reset.	

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

SAF

The SAF name. You can specify one or more SAFs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

SAFAgent

The name of the SAF agent. You can specify one or more SAF agents for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSRemoteEndpoint

The name of the JMS remote endpoint. You can specify one or more JMS remote endpoints for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (FailedMessageTotal is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.

^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JMSRuntime

Use this class for monitoring a WebLogic Java Messaging Service (JMS) service.

The WebLogic_JMSRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
ConnectionsCurrentCount	Returns the current number of connections to this WebLogic Server.	Long
ConnectionsHighCount	Returns the peak number of connections to this WebLogic Server since the last reset.	Long
ConnectionsTotalCount	Returns the total number of connections made to this WebLogic Server since the last reset.	Long

JMSServersCurrentCount	Returns the current number of JMS servers deployed on this WebLogic Server instance.	Long
JMSServersHighCount	Returns the peak number of JMS servers deployed on this WebLogic Server instance since this server was started.	Long
JMSServersTotalCount	Returns the total number of JMS servers deployed on this WebLogic Server instance since this server was started.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMS

The JMS name. You can specify one or more JMSs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (JMSServersCurrentCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.

INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JMSServerRuntime

Use this class for monitoring a WebLogic Java Messaging Service (JMS) server.

The WebLogic_JMSServerRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
BytesCurrentCount	Returns the current number of bytes (excluding pending bytes) stored on this JMS server.	Long
BytesHighCount	Returns the peak number of bytes stored on this JMS server since the last reset.	Long

BytesPageableCurrentCount	Returns the total number of bytes in all the messages that are currently available to be paged out but which have not yet been paged out. The JMS server attempts to keep this number smaller than the "MessageBufferSize" parameter.	Long
BytesPagedInTotalCount	Return the total number of bytes read from the paging directory since the JMS server was started.	Long
BytesPagedOutTotalCount	Returns the total number of bytes written to the paging directory since the JMS server was started.	Long
BytesPendingCount	Returns the current number of bytes pending (unacknowledged or uncommitted) stored on this JMS server. Pending bytes are over and above the current number of bytes.	Long
BytesReceivedCount	Returns the number of bytes received on this JMS server since the last reset.	Long
BytesThresholdTime	Returns the amount of time (in seconds) in the threshold condition since the last reset.	Long
ConsumptionPaused	Returns the current consumption paused state of the JMS server.	Boolean
DestinationsCurrentCount	Returns the current number of destinations for this JMS server.	Long
DestinationsHighCount	Returns the peak number of destinations on this JMS server since the last reset.	Long
DestinationsTotalCount	Returns the total number of destinations instantiated on this JMS server since the last reset.	Long

InsertionPaused	Returns the current insertion paused state of the JMS server as a Boolean value.	Boolean
MessagesCurrentCount	Returns the current number of messages stored on this JMS server. This number does not include pending messages.	Long
MessagesHighCount	Returns the peak number of messages stored on this JMS server since the last reset.	Long
MessagesPageableCurrentCount	Returns the number of messages currently available for paging in this JMS server but not yet paged out. Because of internal implementation details, this count may be zero even if "PageableByteCurrentCount" is zero.	Integer
MessagesPagedInTotalCount	Returns the total number of messages read from the paging directory since the JMS server was started.	Integer
MessagesPagedOutTotalCount	Returns the total number of messages written to the paging directory since the JMS server was started.	Integer
MessagesPendingCount	Returns the current number of messages pending (unacknowledged or uncommitted) stored on this JMS server. Pending messages are over and above the current number of messages.	Long
MessagesReceivedCount	Returns the number of messages received on this destination since the last reset.	Long
MessagesThresholdTime	Returns the amount of time (in seconds) in the threshold condition since the last reset.	Long

ProductionPaused	Returns the current production paused state of the JMS server as a Boolean value.	Boolean
SessionPoolsCurrentCount	Returns the current number of session pools instantiated on this JMS server.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSServer

The name of the JMS server. You can specify one or more JMS servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (BytesCurrentCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.

\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JMSSessionPoolRuntime

Use this class for monitoring a WebLogic Java Messaging Service (JMS) session pool.

The WebLogic_JMSSessionPoolRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
ConnectionConsumersCurrentCount	Returns the current number of connection consumers for this session pool.	Long
ConnectionConsumersHighCount	Returns the peak number of simultaneous connection consumers for this session pool.	Long
ConnectionConsumersTotalCount	Returns the total number of connection consumers made by this session pool since the last reset.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMS

The JMS name. You can specify one or more JMSs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSServers

The name of the JMS server. You can specify one or more JMS servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSSessionPool

The name of the JMS session pool. You can specify one or more JMS session pools for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (ConnectionConsumersHighCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JMSSessionRuntime

Use this class for monitoring a WebLogic Java Messaging Service (JMS) session.

The WebLogic_JMSSessionRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
BytesPendingCount	Returns the number of bytes pending (uncommitted and unacknowledged) for this session.	Long
BytesReceivedCount	Returns the number of bytes received by this session since the last reset.	Long
BytesSentCount	Returns the number of bytes sent by this session since the last reset.	Long
ConsumersCurrentCount	Returns the current number of consumers for this session.	Long
ConsumersHighCount	Returns the peak number of	Long

	consumers for this session since the last reset.	
ConsumersTotalCount	Returns the number of consumers instantiated by this session since the last reset.	Long
MessagesPendingCount	Returns the number of messages pending (uncommitted and unacknowledged) for this session.	Long
MessagesReceivedCount	Returns the number of messages received by this session since the last reset.	Long
MessagesSentCount	Returns the number of bytes sent by this session since the last reset.	Long
ProducersCurrentCount	Returns the current number of producers for this session.	Long
ProducersHighCount	Returns the peak number of producers for this session since the last reset.	Long
ProducersTotalCount	Returns the number of producers for this session since the last reset.	Long
Transacted	Returns whether the session is transacted.	Boolean

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select

between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMS

The JMS name. You can specify one or more JMSs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSConnection

The name of the JMS connection. You can specify one or more JMS connections for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSSession

The name of the JMS session. You can specify one or more JMS sessions for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (BytesSentCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JoltConnectionPoolRuntime

Use this class for monitoring a WebLogic Jolt connection pool.

The WebLogic_JoltConnectionPoolRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
MaxCapacity	Returns the maximum number of connections configured for this Jolt pool.	Integer
SecurityContextPropagation	Indicates whether the security context is propagated.	Boolean

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Jolt

The jolt name. You can specify one or more jolts for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JoltConnectionService

The name of the Jolt connection service. You can specify one or more jolts for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JoltConnectionPool

The name of the Jolt connection pool. You can specify one or more jolts for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JoltConnectionServiceRuntime

Use this class for monitoring a WebLogic Jolt component.

The WebLogic_JoltConnectionServiceRuntime category includes the counter listed in the following table. This counter may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
ConnectionPoolCount	Returns the number of configured Jolt connection pools.	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Jolt

The Jolt name. You can specify one or more Jolts for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JoltConnectionService

The name of the Jolt connection service. You can specify one or more Jolt connection services for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (ConnectionPoolCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.

?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JRockitRuntime

Use this class to expose runtime data about the JRockit virtual machine (VM) that is running the current WebLogic server instance. You cannot change the VM's operating parameters while the VM is active. Instead, use the start-up options described in the JRockit documentation.

The WebLogic_JRockitRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
AllProcessorsAverageLoad	Returns a snapshot of the average load of all processors in the host computer. If the computer has only one processor, this method returns the same value as getJvmProcessorLoad(0). The value is returned as a double, where 1.0 represents 100% load (no idle time), and 0.0 represents 0% load (pure idle time).	Double

Concurrent	Indicates whether the VM's garbage collector (GC) runs in a separate Java thread concurrently with other Java threads.	Boolean
FreeHeap	Returns the amount (in bytes) of Java heap memory currently free in the virtual machine.	Long
FreePhysicalMemory	Returns the amount (in bytes) of physical memory currently free on the host computer.	Long
GcAlgorithm	Indicates the type of GC that the virtual machine is using. JRockit provides the following types of GCs: ! Generational Copying: Suitable for testing applications on a desktop machine with a small (less than 128 MB) heap. ! Single Spaced Concurrent: Reduces or eliminates pauses in the VM that are due to garbage collection. Because it trades memory throughput for reduced pause time, you generally need a larger heap size than with other GC types. If your ordinary Java threads create more garbage than this GC can collect, the VM	Boolean

	will pause while the Java threads wait for the garbage collection to finish. ! Generational Concurrent: Creates a "nursery" space within the heap. New objects are created within the nursery. When the nursery is full, JRockit "stops the world," removes the dead objects from the nursery, and moves live objects to a different space within the heap. Another thread runs in the background to remove dead objects from the non-nursery space. This GC type has a higher memory throughput than a single spaced concurrent GC. ! Parallel: Allocates all objects to a single spaced heap. When the heap is full, all Java threads are stopped and every CPU is used to perform a complete garbage collection of the entire heap. This behavior causes longer pause	

	times than for the concurrent collectors but maximizes memory throughput.	
GCHandlesCompaction	Indicates whether the VM's garbage collector compacts the Java heap. Usually the heap is scattered throughout available memory. A garbage collector that compacts the heap defragments the memory space in addition to deleting unused objects.	Boolean
Generational	Indicates whether the VM's garbage collector uses a nursery space. A nursery is the area of the Java heap that the VM allocates to most objects. Instead of garbage collecting the entire heap, generational garbage collectors focus on the nursery. Because most objects die young, most of the time it is sufficient to garbage collect only the nursery and not the entire heap.	Boolean
HeapFreeCurrent	Returns the current amount of memory (in bytes) that is available in the JVM heap.	Long
HeapFreePercent	Returns the percentage of the maximum memory that is free.	Integer
HeapSizeCurrent	Returns the current size (in bytes) of the JVM heap.	Long
HeapSizeMax	Returns the maximum free memory (in bytes) configured for this JVM.	Long

Incremental	Indicates whether the VM's garbage collector collects (increments) garbage as it scans the memory space and dumps the garbage at the end of its cycle. With a nonincremental garbage collector, garbage is dumped as soon as it is encountered.	Boolean
JvmProcessorLoad	Returns a snapshot of the load that the virtual machine is placing on all processors in the host computer. If the host contains multiple processors, the value represents a snapshot of the average load. The value is returned as a double, where 1.0 represents 100% load (no idle time), and 0.0 represents 0% load (pure idle time).	Double
LastGCEnd	Returns the time at which the last garbage collection run ended.	Long
LastGCStart	Returns the time at which the last garbage collection run started.	Long
NumberOfDaemonThreads	Returns the number of daemon Java threads currently running in the virtual machine across all processors.	Integer
NumberOfProcessors	Returns the number of processors on the virtual machine's host computer. If this is not a Symmetric Multi Processor (SMP) system, the value is 1.	Integer
Parallel	Indicates whether the VM's garbage collector	Boolean

	is able to run in parallel on multiple processors if multiple processors are available.	
TotalGarbageCollectionCount	Returns the number of garbage collection runs having occurred since the virtual machine was started.	Long
TotalGarbageCollectionTime	Returns amount of time (in milliseconds) that the virtual machine has spent on all garbage collection runs since the VM was started.	Long
TotalHeap	Returns the total amount (in bytes) of memory currently allocated to the virtual machine's Java heap. This value, which is also known as the "heap size," may grow up to the specified maximum heap size.	Long
TotalNumberOfThreads	Returns the total number of Java threads (daemon and non-daemon) currently running in the virtual machine across all processors.	Integer
TotalNurserySize	Returns the total amount (in bytes) of memory currently allocated to the nursery, which is the area of the Java heap that the VM allocates to most objects. Instead of garbage collecting the entire heap, generational garbage collectors focus on the nursery. Because most objects die young, most of the time it is sufficient to garbage collect only	Long

	the nursery and not the entire heap. If you are not using a generational garbage collector, the nursery size is 0.	
TotalPhysicalMemory	Returns the total amount (in bytes) of physical memory on the host computer. The value does not include memory that an operating system makes available through swap space on a disk or other types of virtual memory.	Long
Uptime	Returns the amount of time (in milliseconds) that the virtual machine has been running.	Long
UsedHeap	Returns the amount (in bytes) of Java heap memory that is currently being used by the virtual machine.	Long
UsedPhysicalMemory	Returns the amount (in bytes) of physical memory that is currently being used on the host computer. The value describes the memory being used by all processes on the computer, not just by the Virtual Machine.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JRockit

The name of the JRockit virtual machine. You can specify one or more JRockit virtual machines for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (TotalGarbageCollectionCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JTARecoveryRuntime

Use this interface for accessing transaction runtime characteristics for recovered transactions that are associated with a particular transaction recovery service.

The WebLogic_JTARecoveryRuntime (Java Transaction API) category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
Active	Indicates whether the transaction recovery service is currently activated on this server.	Boolean
InitialRecoveredTransactionTotalCount	Returns the total number of transactions recovered from the transaction log initially.	Integer
RecoveredTransactionCompletionPercent	Returns the percentage of transactions recovered from the transaction log initially.	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JTA

The JTA name. You can specify one or more JTAs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JTARecovery

The name of the JTA recovery. You can specify one or more JTA recoveries for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to the counters that are returning a count or total (InitialRecoveredTransactionTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JTARuntime

Use this WebLogic Java Transaction (JTA) API runtime interface for accessing transaction runtime characteristics within a WebLogic server.

The WebLogic_JTARuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
ActiveTransactionsTotalCount	Returns the number of active transactions on the server.	Integer
SecondsActiveTotalCount	Returns the total number of seconds that transactions were active for all committed transactions.	Long
TransactionAbandonedTotalCount	Returns the total number of transactions abandoned since the server was started.	Long
TransactionCommittedTotalCount	Returns the total number of transactions committed since the server was started.	Long
TransactionHeuristicsTotalCount	Returns the number of transactions completed with a heuristic status since the server was started.	Long
TransactionRolledBackAppTotalCount	Returns the number of transactions that were rolled back due to an application error.	Long
TransactionRolledBackResourceTotalCount	Returns the total number of rolled back transactions because of a resource error.	Long
TransactionRolledBackSystemTotalCount	Returns the total number of rolled back transactions because of an internal system	Long

	error.	
TransactionRolledBackTimeoutTotalCount	Returns the total number of rolled back transactions because of a time-out expiration.	Long
TransactionRolledBackTotalCount	Returns the total number of rolled back transactions since the server was started.	Long
TransactionTotalCount	Returns the total number of processed transactions. This total includes all committed, rolled back, and heuristic transaction completions since the server was started.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JTA

The JTA name. You can specify one or more JTAs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (TransactionTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_JVMRuntime

This interface provides methods for retrieving information about the Java Virtual Machine (JVM) within which the current server instance is running. You cannot change the JVM's operating parameters while the JVM is active. Instead, use the startup options described in the JVM's documentation. You may use these methods for any type of JVM that WebLogic Server supports.

The WebLogic_JVMRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
HeapFreeCurrent	Returns the current amount	Long

	of memory (in bytes) available in the JVM heap.	
HeapFreePercent	Returns the percentage of the maximum memory that is free.	Integer
HeapSizeCurrent	Returns the current size (in bytes) of the JVM heap.	Long
HeapSizeMax	Returns the maximum free memory configured for this JVM.	Long
Uptime	Returns the number of milliseconds that the virtual machine has been running.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JVM

The JVM name. You can specify one or more JVMs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_LibraryRuntime

If you use the getMBeanInfo operation in MBeanTypeServiceMBean, supply the following value as this MBean's fully qualified interface name: `weblogic.management.runtime.LibraryRuntimeMBean`

The WebLogic_LibraryRuntime category includes the counter listed in the following table. This counter may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
Referenced	Returns true if this library is referenced by one or more referencers. Typically a library referencer is a deployed application.	Boolean

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter

dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Library

The library name. You can specify one or more libraries for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_LogBroadcasterRuntime

This managed bean (MBean) broadcasts Java Management Extensions (JMX) notifications for each log message generated in the local WLS server. There is exactly one implementation of this MBean in each WLS server. JMX listeners can register to this MBean and receive log notifications. The type of the notification generated is `WebLogicLogNotification`.

The WebLogic_LogBroadcasterRuntime category includes the counter listed in the following table. This counter may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of MBeans in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
MessagesLogged	Returns the total number of log messages that this WebLogic Server instance has generated.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

LogBroadcaster

The name of the log broadcaster. You can specify one or more log broadcasters for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_MaxThreadsConstraintRuntime

This interface provides runtime information for maximum threads constraint.

The WebLogic_MaxThreadsConstraintRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
DeferredRequests	Returns the number of requests that are denied a thread for execution because the constraint is exceeded.	Integer
ExecutingRequests	Returns the number of requests that are currently executing.	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

MaxThreadsConstraint

The name of the maximum threads constraint. You can specify one or more maximum threads constraints for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_MessageDrivenEJBRuntime

This interface contains accessor methods for all Enterprise JavaBean (EJB) runtime information collected for a message-driven bean.

The WebLogic_MessageDrivenEJBRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
JMSConnectionAlive	Returns whether	Boolean

	the message-driven bean is currently connected to the JMS destination to which it is mapped.	
ProcessedMessageCount	Returns the total number of messages processed by this message-driven bean.	Long
SuspendCount	Returns the total number of times this message-driven bean is suspended by the user or the EJB container.	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

EJBComponent

The name of the EJB component. You can specify one or more EJB components for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

MessageDrivenEJB

The name of the message-driven EJB. You can specify one or more message-driven EJBs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (SuspendCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_MinThreadsConstraintRuntime

This interface provides monitoring information for minimum threads constraint.

The WebLogic_MinThreadsConstraintRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type

CompletedRequests	Returns the number of completed requests.	Long
CurrentWaitTime	Returns the last measured time a request had to wait for a thread. Only requests whose execution is needed to satisfy the constraint are considered.	Long
ExecutingRequests	Returns the number of requests that are currently executing.	Integer
MaxWaitTime	Returns the maximum time a request had to wait for a thread. Only requests whose execution is needed to satisfy the constraint are considered.	Long
MustRunCount	Returns the number of requests that must be executed to satisfy the constraint.	Integer
OutOfOrderExecutionCount	Returns the number of requests executed out of turn to satisfy this constraint.	Long
PendingRequests	Returns the pending requests waiting for an available thread.	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select

between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

MinThreadsConstraint

The name of the minimum threads constraint. You can specify one or more minimum threads constraints for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (MustRunCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_NonXAResourceRuntime

Use this interface for accessing runtime statistical information about a non-XA resource.

The WebLogic_NonXAResourceRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
TransactionCommittedTotalCount	Returns the total number of committed transactions since the server was started.	Long
TransactionHeuristicsTotalCount	Returns the number of transactions that completed with a heuristic status since the server was started.	Long
TransactionRolledBackTotalCount	Returns the number of rolled back transactions since the server was started.	Long
TransactionTotalCount	Returns the total number of processed transactions. This total includes all committed, rolled back, and heuristic transaction completions since the server was started.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JTA

The JTA name. You can specify one or more JTAs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

NonXAResource

The name of the non-XA resource. You can specify one or more non-XA resources for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (TransactionTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_PersistentStoreConnectionRuntime

The WebLogic_PersistentStoreConnectionRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
CreateCount	Returns the count of persistent store connections created.	Long
DeleteCount	Returns the count of persistent store connections deleted	Long
ObjectCount	Returns the count of persistent store connections.	Long
ReadCount	Returns the count of persistent store connections read.	Long
UpdateCount	Returns the count of persistent store connections updated.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select

between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

PersistentStore

The name of the persistent store. You can specify one or more persistent store connections for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

PersistentStoreConnection

The name of the persistent store connection. You can specify one or more persistent store connections for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (UpdateCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- ! The primary data point (PDP) is the value returned for that counter.
- ! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_PersistentStoreRuntime

The WebLogic_PersistentStoreRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
CreateCount	Returns the count of persistent stores created.	Long
DeleteCount	Returns the count of persistent stores deleted.	Long
ObjectCount	Returns the count of persistent stores.	Long
PhysicalWriteCount	Returns the count of persistent store physical writes.	Long
ReadCount	Returns the count of persistent stores read.	Long
UpdateCount	Returns the count of persistent stores updated.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter

dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

PersistentStore

The name of the persistent store. You can specify one or more persistent stores for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (UpdateCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_QueryCacheRuntime

This interface contains accessor methods for all query cache runtime information collected for an Enterprise JavaBean (EJB). A query cache miss can occur due to one of five reasons: 1) The query result was not found in the query cache; 2) The query result has timed out; 3) A bean, satisfying the query was not found in the entity cache; 4) A query with relationship-caching turned on did not find the related-beans query result; or, 5) A query which loads multiple EJBs could not load one or more of them. To better aid tuning, there are separate counters provided for each of the last four causes listed above. The fifth counter is a total cache miss counter, which takes into account all five causes of a cache miss.

The WebLogic_QueryCacheRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
CacheAccessCount	Returns the number of accesses of the query cache for this EJB.	Long
CacheHitCount	Returns the number of cache hits of the query cache for this EJB.	Long
CacheMissByBeanEvictionCount	Returns the number of times a cache miss occurred for this EJB because corresponding beans were not found in the entity cache.	Long
CacheMissByDependentQueryMissCount	Returns the number of times a cache miss occurred for this EJB because a dependent query was not found in another EJB's query cache.	Long
CacheMissByRelatedQueryMissCount	Returns the	Long

	number of times a cache miss occurred for this EJB because a related query was not found in another EJB's query cache.	
CacheMissByTimeoutCount	Returns the number of cache misses due to query result time-out for this EJB.	Long
TotalCachedQueriesCount	Returns the total number of query results for this EJB currently in the EJB cache.	Integer
TotalCacheMissCount	Returns the total number of cache misses of the query cache for this EJB.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

EJBComponent

The name of the EJB component. You can specify one or more EJB components for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

EntityEJB

The name of the entity EJB. You can specify one or more entity EJBs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

QueryCache

The name of the query cache. You can specify one or more query caches for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (TotalCacheMissCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_RequestClassRuntime

This interface presents runtime information about request classes, each of which represents a class of work. Work that uses the same request class shares the same priority.

The WebLogic_RequestClassRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the

WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
CompletedCount	Returns the total number of completions since the server was started.	Long
DeltaFirst	Returns the undocumented attribute that exposes a value used in determining priority.	Long
DeltaRepeat	Returns the undocumented attribute that exposes a value used in determining priority.	Long
Interval	Returns the undocumented attribute that exposes a value used in determining priority. This attribute is applicable only for ResponseTimeRequestClass. For FairShareRequestClasses, a -1 is returned.	Double
MyLast	Returns the undocumented attribute that exposes a value used in determining priority.	Long
PendingRequestCount	Returns the number of requests waiting for a thread to become available.	Integer
ThreadUseSquares	Returns the undocumented attribute that exposes a value used in determining priority.	Long
TotalThreadUse	Returns the total amount of thread-use time (in milliseconds) used by the request class since the server was started.	Long
VirtualTimeIncrement	Returns the current priority of the request class. The priority is relative to other request	Long

	class priorities, is dynamically calculated frequently, and can change.	

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WorkManager

The name of the work manager. You can specify one or more work managers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

RequestClass

The name of the request class. You can specify one or more request classes for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (CompletedCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a

	wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_SAFAgentRuntime

Use this class for monitoring a WebLogic Store-and-Forward (SAF) agent.

The WebLogic_SAFAgentRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
BytesCurrentCount	Returns the current number of bytes, excluding pending bytes.	Long
BytesHighCount	Returns the peak number of bytes since the last reset.	Long
BytesPendingCount	Returns the number of pending bytes. Pending bytes are over and above the current number of bytes.	Long
BytesReceivedCount	Returns the	Long

	number of bytes received since the last reset.	
BytesThresholdTime	Returns the amount of time (in seconds) in the threshold condition since the last reset.	Long
ConversationsCurrentCount	Returns the current number of conversations.	Long
ConversationsHighCount	Returns the peak number of conversations since the last reset.	Long
ConversationsTotalCount	Returns the total number of conversations since the last reset.	Long
FailedMessagesTotal	Returns the total number of messages that failed to be forwarded since the last reset.	Long
MessagesCurrentCount	Returns the current number of messages, including pending messages.	Long
MessagesHighCount	Returns the peak number of messages since the last reset.	Long
MessagesPendingCount	Returns the number of pending messages. Pending messages are over and above the current number of messages. A pending message is one sent in a	Long

	transaction and not committed or one forwarded but not acknowledged.	
MessagesReceivedCount	Returns the number of messages received since the last reset.	Long
MessagesThresholdTime	Returns the amount of time (in seconds) in the threshold condition since the last reset.	Long
PausedForForwarding	Indicates whether or not the sending agent is paused for forwarding at the current time.	Boolean
PausedForIncoming	Indicates whether or not the sending agent is paused for incoming messages at the current time.	Boolean
PausedForReceiving	Indicates whether or not the receiving agent is paused for receiving at the current time.	Boolean
RemoteEndpointsCurrentCount	Returns the current number of remote endpoints to which this SAF agent has been storing and forwarding messages.	Long
RemoteEndpointsHighCount	Returns the peak number of remote endpoints to which this SAF agent has been storing and	Long

	forwarding messages since last reset.	
RemoteEndpointsTotalCount	Returns the number of remote endpoints to which this SAF agent has been storing and forwarding messages since last reset.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JMSServer

The name of the Java Messaging Service (JMS) server. You can specify one or more JMS servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

SAFAgent

The name of the SAF agent. You can specify one or more SAF agents for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (FailedMessagesTotal is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_SAFRemoteEndpointRuntime

Use this class for monitoring a WebLogic Store-and-Forward (SAF) remote endpoint.

The WebLogic_SAFRemoteEndpointRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
BytesCurrentCount	Returns the current number of bytes, excluding pending bytes.	Long
BytesHighCount	Returns the peak number of bytes since the last reset.	Long
BytesPendingCount	Returns the number of pending bytes. Pending bytes are	Long

	over and above the current number of bytes.	
BytesReceivedCount	Returns the number of bytes received since the last reset.	Long
BytesThresholdTime	Returns the amount of time (in seconds) in the threshold condition since the last reset.	Long
DowntimeHigh	Returns the longest time (in seconds) that the remote endpoint has not been available since the last reset.	Long
DowntimeTotal	Returns the total time (in seconds) that the remote endpoint has not been available since the last reset.	Long
FailedMessagesTotal	Returns the total number of messages that failed to be forwarded since the last reset.	Long
MessagesCurrentCount	Returns the current number of messages, including pending messages.	Long
MessagesHighCount	Returns the peak number of messages since the last reset.	Long
MessagesPendingCount	Returns the number of pending messages. Pending messages are over and	Long

	above the current number of messages. A pending message is one sent in a transaction and not committed or one forwarded but not acknowledged.	
MessagesReceivedCount	Returns the number of messages received since the last reset.	Long
MessagesThresholdTime	Returns the amount of time (in seconds) in the threshold condition since the last reset.	Long
PausedForForwarding	Indicates whether the remote endpoint is currently not forwarding messages.	Boolean
PausedForIncoming	Indicates whether a remote endpoint is currently not accepting new messages.	Boolean
UptimeHigh	Returns the longest time (in seconds) that the remote endpoint has been available since the last reset.	Long
UptimeTotal	Returns the total time (in seconds) that the remote endpoint has been available since the last reset.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

SAF

The SAF name. You can specify one or more SAFs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

SAFRemoteEndpoint

The name of the SAF remote endpoint. You can specify one or more SAF remote endpoints for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (DowntimeTotal is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_ServerChannelRuntime

Use this class for monitoring runtime information for network access points or "channels."

The WebLogic_ServerChannelRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
AcceptCount	Returns the number of past and present sockets accepted on this channel, thereby providing the connection rate to the server.	Long
BytesReceivedCount	Returns the total number of bytes received on this channel.	Long
BytesSentCount	Returns the total number of bytes sent on this channel.	Long
ConnectionsCount	Returns the number of active connections and sockets associated with this channel.	Long
MessagesReceivedCount	Returns the number of messages received	Long

	on this channel.	
MessagesSentCount	Returns the number of messages sent on this channel.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

ServerChannel

The name of the server channel. You can specify one or more server channels for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (MessagesSentCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_ServerLifeCycleRuntime

This class provides methods that transition servers from one state to another. This class is instantiated only on the administration server, but you can use it to transition the states of managed servers as well as administration servers.

You cannot use this class to start an administration server. If you want to use it to start managed servers, you must first set up a node manager on each managed server host machine.

If you want to use the methods that transition a server into the ADMIN state, you must first set up an administration channel for that server.

The WebLogic_ServerLifeCycleRuntime category includes the counter listed in the following table. This counter may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
NodeManagerRestartCount	Returns the number of times the server has been restarted using the node manager since creation. The first start does not count. The count is valid only if the node manager is used to start and restart the server every time.	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select

between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

ServerLifeCycle

The name of the server life cycle. You can specify one or more server life cycles for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (NodeManagerRestartCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_ServerLifeCycleTaskRuntime

This class exposes monitoring information about a server's life cycle. Remote clients as well as clients running within a server may access this information.

An operation (task) to change a server's state forks a separate thread to perform the actual work and immediately return an instance of this MBean to the caller. The caller may then use this MBean to track the task's progress.

The WebLogic_ServerLifeCycleTaskRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
BeginTime	Returns the task's start time.	Long
EndTime	Returns the task's completion time. A value of -1 indicates that the task is currently running.	Long
Running	Indicates whether the task is still running.	Boolean
SystemTask	Indicates whether this task was initiated by the server or a user.	Boolean

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

ServerLifeCycle

The name of the server life cycle. You can specify one or more server life cycles for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Tasks

The task name. You can specify one or more tasks for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_ServerRuntime

This interface provides methods for retrieving runtime information about a server instance and for transitioning a server from one state to another.

The WebLogic_ServerRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
ActivationTime	Returns the server start time.	Long
AdministrationPort	Returns the port on which this	

	server is listening for administrative request. Deprecated. 9.0.0.0. Replaced by AdministrationURL.	
AdministrationPortEnabled	Indicates whether the administration port is enabled on the server.	Boolean
AdminServer	Indicates whether the server is an administration server.	Boolean
AdminServerListenPort	Returns the port on which the administration server is listening for connections.	Integer
AdminServerListenPortSecure	Indicates whether the port that the server uses for administrative traffic is configured to use a secure protocol.	Boolean
ListenPort	Returns the port on which this server is listening for connections. Deprecated. 9.0.0.0. Replaced by URL.	
ListenPortEnabled	Indicates whether the default listen port is enabled on the server.	Boolean
OpenSocketsCurrentCount	Returns the current number of sockets registered for socket muxing on this server.	Integer
RestartRequired	Indicates whether the server must be restarted in order to activate configuration	Boolean

	changes.	
RestartsTotalCount	Returns the total number of restarts for this server since the cluster was last started.	Integer
SocketsOpenedTotalCount	Returns the total number of registrations for socket muxing on this sever.	Long
SSLListenPort	Returns the port on which this server is listening for SSL connections. Deprecated. 9.0.0.0. Replaced by URL.	Integer
SSLListenPortEnabled	Indicates whether the default SSL listen port is enabled on the server.	Boolean
StateVal	Returns the current state of the server as an integer.	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to counters returning a count or total (SocketsOpenedTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw data value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_ServerSecurityRuntime

Use this class for monitoring WebLogic security information.

The WebLogic_ServerSecurityRuntime category includes the counter listed in the following table. This counter may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
JACCEnabled	Indicates whether Java Authorization Contract for Containers	Boolean

	(JACC) was enabled on the command line for the Java Virtual Machine hosting this server.	

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

ServerSecurity

The name of the server security. You can specify one or more server securities for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_ServletRuntime

This interface describes servlet activity.

The WebLogic_ServletRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
ExecutionTimeAverage	Returns the average amount of time all invocations of the servlet have executed since created.	Integer
ExecutionTimeHigh	Returns the amount of time the single longest invocation of the servlet has executed since created.	Integer
ExecutionTimeLow	Returns the amount of time the single shortest invocation of the servlet has executed since created.	Integer
ExecutionTimeTotal	Returns the total amount of time all invocations of the servlet has executed since created.	Integer
InvocationTotalCount	Returns a total count of the times this servlet has been	Integer

	invoked.	
PoolMaxCapacity	Returns the maximum capacity of this servlet for single thread model servlets.	Integer
ReloadTotalCount	Returns the total count of the number of times this servlet has been reloaded.	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WebAppComponent

The name of the Web application component. You can specify one or more Web application components for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Servlet

Servlet name. You can specify one or more servlets for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to the counters that are returning a count or total (ExecutionTimeTotal is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_TaskRuntime

This interface exposes monitoring information about an ongoing and potentially long-running administrative process. This means, at minimum, any OA&M operation involving I/O. Examples include starting and stopping servers, deploying and undeploying applications, or migrating services.

A managed bean (MBean) operation of this sort should fork a separate thread to perform the actual work and immediately return an instance of TaskRuntimeMBean to the caller. The caller can then use this to track the task's progress as desired. Users can also query for all instances of TaskRuntimeMBean to get a summary of both currently running and recently completed tasks.

An instance of TaskRuntimeMBean continues to exist in the MBeanServer after the completion of the work they describe. They will eventually either be explicitly deregistered by the user or removed by a scavenger process, which periodically purges TaskRuntimeMBeans that have been completed for some time.

The WebLogic_TaskRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
BeginTime	Returns the start time for this task.	Long
EndTime	Returns the completion time for this task. A value of -1 means that the task is currently running.	Long
Running	Indicates whether the task is still running	Boolean
SystemTask	Indicates whether this task was initiated by the server or a user.	Boolean

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

ServerLifeCycle

The name of the server life cycle. You can specify one or more server life cycles for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Tasks

The task name. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

SubTasks

The name of the subtask. You can specify one or more subtasks for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.

?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- ! The primary data point (PDP) is the value returned for that counter.
- ! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_ThreadPoolRuntime

Use this bean to monitor the self-tuning queue.

The WebLogic_ThreadPoolRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
CompletedRequestCount	Returns the number of completed requests in the priority queue.	Long
ExecuteThreadIdleCount	Returns the number of idle threads in the pool.	Integer
ExecuteThreadTotalCount	Returns the total number of threads in the pool.	Integer
HoggingThreadCount	Returns the threads that are being hogged by	Integer

	a request right now. These threads will either be declared as stuck after the configured timeout or will return to the pool before that. The self-tuning mechanism will backfill if necessary.	
MinThreadsConstraintsCompleted	Returns the number of requests with minimum threads constraint picked up out of order for execution immediately since their min threads requirement was not met. This number does not include a situation where threads are idle during schedule.	Long
MinThreadsConstraintsPending	Returns the number of requests that should be executed now to satisfy the minimum threads requirement.	Integer
PendingUserRequestCount	Returns the number of pending user requests in the priority queue. The priority queue contains requests from internal subsystems and users. This number represents the	Integer

	count of all user requests.	
QueueLength	Returns the number of pending requests in the priority queue. This is the total of internal system requests and user requests.	Integer
SharedCapacityForWorkManagers	Returns the maximum amount of requests that can be accepted in the priority queue. A request with higher priority is accepted in place of a lower-priority request already in the queue even after the threshold is reached. The lower priority request is kept waiting in the queue until all high-priority requests are executed. Further enqueues of the low priority requests are rejected immediately.	Integer
StandbyThreadCount	Returns the number of threads in the standby pool. Surplus threads that are not needed to handle the present workload are designated as standby and added to the standby pool. These threads are activated when	Integer

	more threads are needed.	
Suspended	Indicates if the RequestManager is suspended. A suspended manager will not dequeue work and dispatch threads until it is resumed.	Boolean
Throughput	Returns the mean number of requests completed per second.	Double

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

ThreadPool

The name of the thread pool. You can specify one or more thread pools for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to the counters that are returning a count or total (ExecuteThreadTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_TransactionNameRuntime

This interface represents runtime statistics for a transaction name category.

The WebLogic_TransactionNameRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
SecondsActiveTotalCount	Returns the total number of seconds for all committed, active transactions.	Long
TransactionAbandonedTotalCount	Returns the total number of abandoned transactions since the server was started.	Long
TransactionCommittedTotalCount	Returns the total	Long

	number of committed transactions since the server was started.	
TransactionHeuristicsTotalCount	Returns the total number of completed transactions with a heuristic status since the server was started.	Long
TransactionRolledBackAppTotalCount	Returns the total number of rolled-back transactions because of an application error.	Long
TransactionRolledBackResourceTotalCount	Returns the total number of rolled back transactions because of a resource error.	Long
TransactionRolledBackSystemTotalCount	Returns the number of rolled-back transaction because of an internal system error.	Long
TransactionRolledBackTimeoutTotalCount	Returns the number of rolled-back transactions because of a timeout expiration.	Long
TransactionRolledBackTotalCount	Returns the number of rolled-back transactions since the server was started.	Long
TransactionTotalCount	Returns the total number of processed transactions, including all committed, rolled-back, and heuristic transaction completions since the server	Long

	was started.	

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JTA

The JTA name. You can specify one or more JTAs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

TransactionResource

The name of the transaction resource. You can specify one or more transaction resources for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to the counters that are returning a count or total (SecondsActiveTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.

^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_TransactionResourceRuntime

This interface represents runtime statistics for a transactional resource.

The WebLogic_TransactionResourceRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
TransactionCommittedTotalCount	Returns the total number of transactions committed since the server was started.	Long
TransactionHeuristicCommitTotalCount	Returns the total number of transactions for which this resource has returned a heuristic commit decision.	Long
TransactionHeuristicHazardTotalCount	Returns the total number of transactions for which this resource has reported a	Long

	heuristic hazard decision.	
TransactionHeuristicMixedTotalCount	Returns the total number of transactions for which this resource has reported a heuristic mixed decision.	Long
TransactionHeuristicRollbackTotalCount	Returns the total number of transactions for which this resource has returned a heuristic rollback decision.	Long
TransactionHeuristicsTotalCount	Returns the total number of transactions that completed with a heuristic status since the server was started.	Long
TransactionRolledBackTotalCount	Returns the total number of transactions that were rolled back since the server was started.	Long
TransactionTotalCount	Returns the total number of processed transactions, including all committed, rolled back, and heuristic transaction completions since the server was started.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

JTA

The Java Transaction API (JTA) name. You can specify one or more JTAs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

TransactionResource

The name of the transaction resource. You can specify one or more transaction resources for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to the counters that are returning a count or total (TransactionTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_UserLockoutManagerRuntime

Use this class to monitor and manage per security realm user lockout information.

The WebLogic_UserLockoutManagerRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
InvalidLoginAttemptsTotalCount	Returns the total number of invalid logins attempted since this server started and lockouts enabled. In a cluster, this method returns the number of invalid logins attempted that have occurred since the cluster started because all servers share login failure information.	Long
InvalidLoginUsersHighCount	Returns the highest number of users with concurrent unexpired or uncleared invalid login attempts. Invalid log in attempts expire as specified by LockoutResetDuration. Use this count to determine whether you need to modify the LockoutCacheSize.	Long
LockedUsersCurrentCount	Returns the number of users currently locked out of this server.	Long

LoginAttemptsWhileLockedTotalCount	Returns the total number of invalid logins attempted since this server started and lockouts enabled.	Long
UnlockedUsersTotalCount	Returns the total number times users have been unlocked since this server started.	Long
UserLockoutTotalCount	Returns the total number of user lockouts that occurred since this server started. In a cluster, this method returns the number of user lockouts that occurred since the cluster started because all servers share login failure information.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

ServerSecurity

The name of the server security. You can specify one or more server securities for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Realm

The realm name. You can specify one or more realms for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

UserLockoutManager

The name of the user lockout manager. You can specify one or more user lockout managers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to the counters that are returning a count or total (UserLockoutTotalCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_WANReplicationRuntime

The WebLogic_WANReplicationRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type

NumberOfSessionsFlushedToTheDatabase	Returns the number of sessions emptied to the database.	Long
NumberOfSessionsRetrievedFromTheDatabase	Returns the number of sessions called up from the database.	Long
PrimaryCount	Returns the number of objects that the local server hosts as primaries.	Long
RemoteClusterReachable	Indicates whether the remote cluster is reachable.	Boolean
SecondaryCount	Returns the number of objects that the local server hosts as secondaries.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WANReplication

The name of the WAN replication. You can specify one or more WAN replications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to the counters that are returning a count or total (SecondaryCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_WebAppComponentRuntime

This interface describes a servlet component (servlet context).

The WebLogic_WebAppComponentRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
DeploymentState	Returns the current deployment state of the module.	Integer

FilterDispatchedRequestsEnabled	Indicates whether the dispatched requests are filtered as configured in weblogic.xml.	Boolean
IndexDirectoryEnabled	Indicates whether the current directory indexing indicator is configured in weblogic.xml.	Boolean
JSPDebug	Returns the Java Server Page debug or line numbers parameter value as it is configured in weblogic.xml.	Boolean
JSPKeepGenerated	Returns the Java Server Page keep generated parameter value as it is configured in weblogic.xml.	Boolean
JSPPageCheckSecs	Returns the Java Server Page page check seconds as it is configured in weblogic.xml.	Long
JSPVerbose	Returns the Java Server Page verbose parameter value as it is configured in weblogic.xml.	Boolean
OpenSessionsCurrentCount	Returns the current total number of open sessions in this component.	Integer
OpenSessionsHighCount	Returns the peak watermark of the total number of open sessions in this server. The count starts at zero each time the server is	Integer

	activated. This is an optimization method for a highly useful statistic that could be implemented less efficiently using change notification.	
ServletReloadCheckSecs	Returns the servlet reload check seconds as it is configured in weblogic.xml.	Integer
SessionCookieMaxAgeSecs	Returns the life span of the session cookie (in seconds) after which it expires on the client. If the value is zero, the cookie expires immediately. If set to -1, the cookie expires when the user exits the browser.	Integer
SessionIDLength	Returns the session identifier length configured for HTTP sessions.	Integer
SessionInvalidationIntervalSecs	Returns the invalidation check timer interval configured for HTTP sessions.	Integer
SessionMonitoringEnabled	Returns the session monitoring indicator as it is configured in weblogic.xml.	Boolean
SessionsOpenedTotalCount	Returns a count of the total number of sessions opened.	Integer

SessionTimeoutSecs	Returns the timeout configured for HTTP sessions.	Integer
SingleThreadedServletPoolSize	Returns the single threaded servlet pool size as it is configured in weblogic.xml.	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WebAppComponent

The name of the Web application component. You can specify one or more Web application components for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to the counters that are returning a count or total (OpenSessionsHighCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_WebServerRuntime

This interface describes a Web server (HTTP server).

The WebLogic_WebServerRuntime category includes the counter listed in the following table. This counter may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
DefaultWebServer	Indicates whether the server is the default Web server or a virtual host.	Boolean

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WebServer

The name of the Web server. You can specify one or more Web servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- ! The primary data point (PDP) is the value returned for that counter.
- ! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_WLDFArchiveRuntime

Use this interface to collect statistical information about the data archives maintained by WLDF. Information provided by this interface is common to all WLDF data archives.

The WebLogic_WLDFArchiveRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
RecordRetrievalTime	Returns the time (in milliseconds) spent retrieving records from the archive since the server was started.	Long
RecordSeekCount	Returns the number of seek operations performed on the archive since the server was started.	Long
RecordSeekTime	Returns the time (in milliseconds) spent locating the first record during a query operation since the server was started.	Long
RetrievedRecordCount	Returns the number of records retrieved from the archive since the server was started.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WLDF

The WLDF name. You can specify one or more WLDFs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WLDFArchive

The WLDF name. You can specify one or more WLDFs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to the counters that are returning a count or total (RecordSeekCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_WLDFDataAccessRuntime

Use this interface to access the specific type of diagnostic data from an underlying log for which this instance is created.

The WebLogic_WLDFDataAccessRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
EarliestAvailableTimestamp	Returns the timestamp (in milliseconds) since Jan 1, 1970 AD, 00:00:00 GMT for the earliest record in the diagnostic data log.	Long
LatestAvailableTimestamp	Returns the timestamp (in milliseconds) since Jan 1, 1970 AD, 00:00:00 GMT for the newest record in the diagnostic data log.	Long
LatestRecordId	Returns the latest known record identifier for the underlying archive.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WLDF

The WLDF name. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WLDFAccess

The name of the WLDF access. You can specify one or more WLDF accesses for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WLDFDataAccess

The name of the WLDF data access. You can specify one or more WLDF data accesses for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_WLDFDbstoreArchiveRuntime

Use this interface to retrieve statistical information associated with the WLDF archives that use databases for storage.

The WebLogic_WLDFDbstoreArchiveRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
DeletionCount	Returns the number of records deleted from the archive	Long

	since this server was started.	
DeletionTime	Returns the time (in milliseconds) spent deleting records from the archive since this server was started.	Long
InsertionCount	Returns the number of new records inserted in the archive since this server was started.	Long
InsertionTime	Returns the time (in milliseconds) spent inserting records into the archive since this server was started.	Long
RecordRetrievalTime	Returns the time (in milliseconds) spent retrieving records from the archive since the server was started.	Long
RecordSeekCount	Returns the number of seek operations performed on the archive since the server was started.	Long
RecordSeekTime	Returns the time (in milliseconds) spent locating the first record during a query operation since the server was started.	Long
RetrievedRecordCount	Returns the number of records retrieved from the archive since the server	Long

	was started.	

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WLDF

The WLDF name. You can specify one or more WLDFs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WLDFDbstoreArchive

The name of the WLDF database store archive. You can specify one or more WLDF database store archives for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to the counters that are returning a count or total (RecordSeekCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_WLDFFileArchiveRuntime

Use this interface to collect statistical information about file-based WebLogic Diagnostic Framework (WLDF) archives.

The WebLogic_WLDFFileArchiveRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
IncrementalIndexCycleCount	Returns the number of times incremental indexing cycles were executed since the server was started.	Integer
IncrementalIndexTime	Returns the cumulative time (in milliseconds) spent performing incremental indexing since the server was started.	Long
IndexCycleCount	Returns the number of times indexing cycles were executed since the server was started.	Integer
IndexTime	Returns the cumulative	Long

	indexing time (in milliseconds) since the server was started.	
RecordRetrievalTime	Returns the time (in milliseconds)spent retrieving records from the archive since the server was started.	Long
RecordSeekCount	Returns the number of seek operations performed on the archive since the server was started.	Long
RecordSeekTime	Returns the time (in milliseconds) spent locating the first record during a query operation since the server was started.	Long
RetrievedRecordCount	Returns the number of records retrieved from the archive since the server was started.	Long
RotatedFilesCount	Returns the number of rotated log file fragments.	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WLDF

The WLDF name. You can specify one or more WLDFs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WLDFFileArchive

The name of the WLDF file archive. You can specify one or more WLDF file archives for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to the counters that are returning a count or total (RotatedFilesCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_WLDFHarvesterRuntime

The WebLogic Diagnostic Framework (WLDF) category provides information about harvestable and harvested attributes, types, and instances. "Harvestable" means potentially available for harvesting; "harvested" means explicitly designated for harvesting. These terms apply to types, instances, and the attributes within those types. In addition, the interface provides access to sampling and snapshot statistics. All statistics are based on data collected during the current server session.

The WebLogic WLDFHarvesterRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
AverageSamplingTime	Returns the average amount of time (in nanoseconds) spent in sampling cycles.	Long
CurrentDataSampleCount	Returns the number of collected data samples in the current snapshot.	Long
CurrentSampleTimeAnOutlier	Indicates whether the sampling time for the most recent data sample differed significantly enough from the average to be considered a statistical outlier.	Boolean
CurrentSnapshotElapsedTime	Returns the elapsed time (in nanoseconds) of a snapshot.	Long
CurrentSnapshotStartTime	Returns the start time (in nanoseconds) of a snapshot.	Long
MaximumSamplingTime	Returns the maximum sampling time (in	Long

	nanoseconds).	
MinimumSamplingTime	Returns the minimum sampling time (in nanoseconds).	Long
OutlierDetectionFactor	Returns the multiplicative factor used to determine a statistical outlier. If the actual sampling time exceeds this, the session average multiplied by the outlier detection factor, then the sampling time is considered a statistical outlier.	Float
SamplePeriod	Returns the current global sample period (in nanoseconds).	Long
TotalDataSampleCount	Returns the number of configured data samples collected so far in this server session.	Long
TotalSamplingCycles	Returns the total number of sampling cycles taken so far.	Long
TotalSamplingTime	Returns the total amount of time (in nanoseconds) spent in sampling cycles.	Long
TotalSamplingTimeOutlierCount	Returns the number of times within this server session that the sampling time differed significantly enough from the average to be considered a	Long

	statistical outlier. The harvester removes these values from the ongoing averages.	

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WLDF

The WLDF name. You can specify one or more WLDFs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WLDFHarvester

The name of WLDF harvester. You can specify one or more WLDF harvesters for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to the counters that are returning a count or total (TotalDataSampleCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.

\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_WLDFImageCreationTaskRuntime

This interface exposes monitoring information about a potentially long-running request for the generation of a diagnostic image. Remote clients, as well as clients running within a server, can access this information.

WebLogic Diagnostic Framework (WLDF) Image Runtime supports operations to request the generation of a diagnostic image for capturing a running server's internal state information. These operations will fork a separate thread to perform the actual work and immediately return an instance of this MBean to the caller. The caller can then use that instance to track the task's progress.

The WebLogic_WLDFImageCreationTaskRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
BeginTime	Returns the task start time.	Long
EndTime	Returns the task completion time. A value of -1 indicates that the task is currently running.	Long
Running	Indicates whether the task is still running.	Boolean
SystemTask	Indicates whether this task was initiated by the	Boolean

	server versus a user.	

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Location

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_WLDFInstrumentationRuntime

This WebLogic DiagnosticFramework (WLDF) interface defines various methods for accessing runtime information about the diagnostic instrumentation system.

The WebLogic_WLDFInstrumentationRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX)Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
CallJoinpointCount	Returns the number of affected CALL join points for all inspected classes. (CALL join points are on the caller side.)	Integer
ExecutionJoinpointCount	Returns the number of affected EXECUTION join points for all inspected classes. (EXECUTION join points are on the callee side.)	Integer
InspectedClassesCount	Returns the number of classes inspected for weaving. (Weaving is the insertion of diagnostic code.)	Integer
MaxWeavingTime	For all classes, returns the weaving time (in nanoseconds) for the class that required the most time to process, including the time spent for inspection and for modification.	Long
MinWeavingTime	For all classes, returns the weaving time (in nanoseconds) for the class that	Long

	required the least time to process, including the time spent for inspection and for modification.	
ModifiedClassesCount	Returns the number of modified classes (i.e., classes where diagnostic code has been inserted).	Integer
TotalWeavingTime	For all classes, returns the total weaving time (in nanoseconds) for processing, including the time spent for inspection and for modification.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WLDF

The WLDF name. You can specify one or more WLDFs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WLDFInstrumentation

The name of the WLDF instrumentation. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to the counters that are returning a count or total (ModifiedClassesCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_WLDFWatchNotificationRuntime

This WebLogic Diagnostic Framework (WLDF) Watch Notification Runtime interface provides access to watch and notification statistical data for the current instance of this server.

The WebLogic_WLDFWatchNotificationRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
AverageEventDataWatchEvaluationTime	Returns the average instrumentation event data	Long

	evaluation cycle time (in milliseconds).	
AverageHarvesterWatchEvaluationTime	Returns the average harvester evaluation cycle time (in milliseconds).	Long
AverageLogWatchEvaluationTime	Returns the average log evaluation cycle time (in milliseconds).	Long
CurrentActiveAlarmsCount	Returns the number of active alarms of any type.	Integer
MaximumActiveAlarmsCount	Returns the maximum number of active alarms at any one time.	Integer
MaximumEventDataWatchEvaluationTime	Returns the maximum time spent evaluating event data watches.	Long
MaximumHarvesterWatchEvaluationTime	Returns the maximum time spent evaluating harvester watches.	Long
MaximumLogWatchEvaluationTime	Returns the maximum time spent evaluating log watches.	Long
MinimumEventDataWatchEvaluationTime	Returns the minimum time spent evaluating log watches.	Long
MinimumHarvesterWatchEvaluationTime	Returns the minimum time spent evaluating harvester watches.	Long
MinimumLogWatchEvaluationTime	Returns the minimum time	Long

	spent evaluating log watches.	
TotalActiveAutomaticResetAlarms	Returns the total number of active automatically reset alarms.	Long
TotalActiveManualResetAlarms	Returns the total number of active manually reset alarms.	Long
TotalDIMGNotificationsPerformed	Returns the total number of diagnostic image notifications fired. Diagnostic image files are not true notifications, but this component records the number of image captures requested by the watch component.	Long
TotalEventDataEvaluationCycles	Returns the total number of times instrumentation event data watch rules have been evaluated.	Long
TotalEventDataWatchesTriggered	Returns the total number of instrumentation event data watch rules that evaluated to true and triggered notifications.	Long
TotalEventDataWatchEvaluations	Returns the total number of evaluated instrumentation event data watch rules. For each cycle, the watch and notification component evaluates all of the enabled	Long

	instrumentation event data watches.	
TotalFailedDIMGNotifications	Returns the total number of failed diagnostic image notification requests.	Long
TotalFailedJMSNotifications	Returns the total number of failed Java Message Service (JMS) notification attempts.	Long
TotalFailedJMXNotifications	Returns the total number of failed JMX notification attempts.	Long
TotalFailedNotifications	Returns the total number of failed notification requests.	Long
TotalFailedSMTPNotifications	Returns the total number of failed Simple Mail Transfer Protocol (SMTP) notification attempts.	Long
TotalFailedSNMPNotifications	Returns the total number of failed Simple Network Management Protocol (SNMP) notification attempts.	Long
TotalHarvesterEvaluationCycles	Returns the total number of times the harvester invoked the watch and notification component to evaluate harvester watch rules. (This number corresponds to the number of	Long

	sampling cycles.)	
TotalHarvesterWatchesTriggered	Returns the total number of harvester watch rules that evaluated to true and triggered notifications.	Long
TotalHarvesterWatchEvaluations	Returns the total number of evaluated harvester watch rules. For each cycle, the watch and notification component evaluates all of the enabled harvester watches.	Long
TotalJMSNotificationsPerformed	Returns the total number of Java Message Service (JMS) notifications successfully fired.	Long
TotalJMXNotificationsPerformed	Returns the total number of JMX notifications successfully fired.	Long
TotalLogEvaluationCycles	Returns the total number of times log watch rules have been evaluated.	Long
TotalLogWatchesTriggered	Returns the total number of log watch rules that evaluated to true and triggered notifications.	Long
TotalLogWatchEvaluations	Returns the total number of evaluated log watch rules. For each cycle, the watch and notification component	Long

	evaluates all of the enabled log watches.	
TotalNotificationsPerformed	Returns the total number of notifications performed.	Long
TotalSMTPNotificationsPerformed	Returns the total number of SMTP notifications successfully fired.	Long
TotalSNMPNotificationsPerformed	Returns the total number of Simple Network Management Protocol (SNMP) notifications successfully fired.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WLDF

The WLDF name. You can specify one or more WLDFs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WLDFWatchNotification

The name of the WLDF watch notification. You can specify one or more WLDF watch notifications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to the counters that are returning a count or total (CurrentActiveAlarmsCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_WLDFWlstoreArchiveRuntime

Use this interface to retrieve statistical information associated with WebLogic Diagnostic Framework (WLDF) archives that use WebLogic Store for data storage.

The WebLogic_WLDFWlstoreArchiveRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
DeletionCount	Returns the number of records deleted since the server was started.	Long

DeletionTime	Returns the cumulative time (in milliseconds) spent to delete records since the server was started.	Long
IndexPageCount	Returns the number of index pages.	Integer
InsertionCount	Returns the number of records created since the server was started.	Long
InsertionTime	Returns the cumulative time (in milliseconds) spent to insert records since the server was started.	Long
RecordRetrievalTime	Returns the time (in milliseconds) spent retrieving records from the archive since the server was started.	Long
RecordSeekCount	Returns the number of seek operations performed on the archive since the server was started.	Long
RecordSeekTime	Returns the time (in milliseconds) spent locating the first record during a query operation since the server was started.	Long
RetrievedRecordCount	Returns the number of records retrieved from the archive since the server	Long

	was started.	

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to the counters that are returning a count or total (DeletionCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_WorkManagerRuntime

Use this interface for monitoring work manager runtime information.

The WebLogic_WorkManagerRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
CompletedRequests	Returns the number of processed requests.	Long
PendingRequests	Returns the number of waiting requests in the queue.	Integer
StuckThreadCount	Returns the number of stuck threads based on any stuck thread constraints.	Integer

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WorkManager

Name of the work manager. You can specify one or more work managers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to the counters that are returning a count or total (StuckThreadCount is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic Wsee Operation Runtime

This interface describes the state of a particular Web service operation, such as deployment state and runtime statistics about the execution of the operation.

The WebLogic_WseeOperationRuntimeOperationRuntime (Web Services Execution Engine) category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
DispatchTimeAverage	Returns the average dispatch time for the current measurement period. Dispatch time refers to the time for WebLogic server to process the invocation. The measurement period typically starts when the WebLogic Server was first started.	Long
DispatchTimeHigh	Returns the longest dispatch time for the current measurement period. Dispatch time refers to the time for the WebLogic Server to process the invocation. The measurement period typically starts when the WebLogic server was first started.	Long
DispatchTimeLow	Returns the lowest dispatch time for the current measurement period. Dispatch time refers to the time for the WebLogic Server to process the invocation. The measurement period typically starts when the WebLogic server was first started.	Long
DispatchTimeTotal	Returns the total time for all	Long

	dispatches of this operation in the current measurement period. Dispatches refer to the time for the WebLogic Server to process the invocation. The measurement period typically starts when the WebLogic Server was first started.	
ExecutionTimeAverage	Returns the average execution time of this operation.	Long
ExecutionTimeHigh	Returns the longest execution time of this operation.	Long
ExecutionTimeLow	Returns the lowest execution time of this operation.	Long
ExecutionTimeTotal	Returns the total time for all executions of this operation.	Long
InvocationCount	Returns the total number of times that this operation has been invoked in the current measurement period. The measurement period typically starts when the WebLogic server was first started.	Integer
ResponseCount	Returns the total number of responses generated from invocations of	Integer

	this operation.	
ResponseErrorCount	Returns the total number of errors from responses generated from invocations of this operation.	Integer
ResponseTimeAverage	Returns the average response time from the responses generated from invocations of this operation.	Long
ResponseTimeHigh	Returns the longest response time based on responses generated from invocations of this operation.	Long
ResponseTimeLow	Returns the lowest response time based on responses generated from invocations of this operation.	Long
ResponseTimeTotal	Returns the total time for all responses generated from invocations of this operation.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Application

The application name. You can specify one or more applications for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Wsee

The Web Services Execution Engine (Wsee) name. You can specify one or more Wsees for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WseePort

The name of the Wsee port. You can specify one or more Wsee ports for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WseeOperation

The name of the Wsee operation. You can specify one or more Wsee operations for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to the counters that are returning a count or total (ResponseTimeTotal is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

! The primary data point (PDP) is the value returned for that counter.

! The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebLogic_WSRMRemoteEndpointRuntime

Use this class for monitoring a WebLogic Store-and-Forward (SAF) remote endpoint for Web services reliable messaging.

The WebLogic_WSRMRemoteEndpointRuntime category includes the counters listed in the following table. Some of these counters may not be available on your system. ServerVantage dynamically discovers WebLogic counter categories, counter names, and parameters by processing the set of managed beans (MBeans) in the WebLogic Java Management Extensions (JMX) Server. Which counters are discovered is determined by the WebLogic version you are running and how WebLogic is configured.

Counters

Counter	Description	Data Point Type
BytesCurrentCount	Returns the current number of bytes, excluding pending bytes.	Long
BytesHighCount	Returns the peak number of bytes since the last reset.	Long
BytesPendingCount	Returns the number of pending bytes. Pending bytes are over and above the current number of bytes.	Long
BytesReceivedCount	Returns the number of bytes received since the last reset.	Long
BytesThresholdTime	Returns the amount of time in the threshold condition since the last reset.	Long
ConversationsCurrentCount	Returns the current number of conversations.	Long

ConversationsHighCount	Returns the peak number of conversations since the last reset.	Long
ConversationsTotalCount	Returns the total number of conversations since the last reset.	Long
DowntimeHigh	Specifies the longest time (in seconds) that the remote endpoint has not been available since the last reset.	Long
DowntimeTotal	Specifies the total time (in seconds) that the remote endpoint has not been available since the last reset.	Long
FailedMessagesTotal	Returns the total number of failed messages to be forwarded since the last reset.	Long
MessagesCurrentCount	Returns the current number of messages, including pending messages.	Long
MessagesHighCount	Returns the peak number of messages since the last reset.	Long
MessagesPendingCount	Returns the number of pending messages, which are over and above the current number of messages. A pending message is one sent in a	Long

	transaction and not committed or forwarded but not acknowledged.	
MessagesReceivedCount	Returns the number of messages received since the last reset.	Long
MessagesThresholdTime	Returns the amount of time in the threshold condition since the last reset.	Long
PausedForForwarding	Indicates if the remote endpoint is currently not forwarding messages.	Boolean
PausedForIncoming	Indicates if a remote endpoint is currently not accepting new messages.	Boolean
UptimeHigh	Returns the longest time (in seconds) that the remote endpoint has been available since the last reset.	Long
UptimeTotal	Returns the total time (in seconds) that the remote endpoint has been available since the last reset.	Long

Parameters

The parameters for a WebLogic counter category are derived from the MBean name. The parameters values are analyzed and displayed according to their parameter dependency structure. This allows you to select between multiple parameters and always end up with a valid combination of parameters. This parameter dependency information is enforced by the task creation wizard in the VantageView Web Console Management function.

The following parameters are valid for this counter category.

Domain

Domain in which the WebLogic Application Admin server and its managed servers reside. You can specify one or more domains for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Server

WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

SAF

The SAF name. You can specify one or more SAFs for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

SAFAgent

The name of the SAF agent. You can specify one or more SAF agents for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

WSRMRemoteEndpoint

The name of the WSRM remote endpoint. You can specify one or more WSRM remote endpoints for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

StatType

This parameter applies only to the counters that are returning a count or total (FailedMessagesTotal is one example in this counter category). Possible values are:

ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

Valid Wildcard Characters

For parameters that can be defined with wildcard patterns, the valid wildcard characters are:

*	Asterisk, represents zero or more characters as a wildcard.
?	Question mark, represents any one individual character as a wildcard.
^	Caret, excludes all values that match the specified pattern.
\	Backslash, precede a wildcard character with a backslash to represent the wildcard character as part of the actual value.

For more information, see Entering Wildcard Parameters.

Data Point

For each counter that you have included in a task:

- The primary data point (PDP) is the value returned for that counter.
- The intelligent data point (IDP) is the set of values returned for all counters in the counter category.

The data point type and the parameters specified in the task determine your data point. See WebLogic Data Points for detail.

Interval

Recommended minimum is 5 minutes.

WebSphere Counters

WebSphere Remote Extended Counters

The following dynamically discovered WebSphere remote extended counter categories are provided in QALoad. Each category provides counters that extend the monitoring of your WebSphere system. The categories, counter names, and parameters are all dynamically discovered by processing data available from the WebSphere Performance Monitoring Infrastructure.

Remote monitoring supports WebSphere versions: 4.0+, 5.0, and 6.0. The counters supported vary by version.

WebSphere Alarm Manager Counters	WebSphere ORB Perf Module
WebSphere Bean Module	WebSphere Scheduler Module
WebSphere Cache Module	WebSphere Servlet Sessions Module
WebSphere Connection Pool Module	WebSphere System Module
WebSphere DCS Stack Counters	WebSphere Thread Pool Module
WebSphere High Availability Manager Counters	WebSphere Transaction Module
WebSphere J2C Module	WebSphere Web App Module
WebSphere JVM Runtime Module	WebSphere Web Services Counters

WebSphere Alarm Manager Counters

The counters discovered for the WebSphere Alarm Manger category are determined by the level of metrics you set in WebSphere. The WebSphere Alarm Manager data counters may include the following counters:

Counter Name (6.0 and greater)	Counter Name (5.0 or earlier)	Description	WebSphere Version	Level of Metrics	Data Point Type
AlarmsCreatedCount		Total number of alarms created by all asynchronous scopes for this .WorkManager.	5.0 and above	High	Long
AlarmsCancelledCount		Number of alarms cancelled by the application.	5.0 and above	High	Long
AlarmsFiredCount		Number of alarms fired.	5.0 and above	High	Long

AlarmLatencyDuration		Latency of alarms fired in milliseconds.	5.0 and above	High	Load
AlarmsPendingSize		Number of alarms waiting to fire.	5.0 and above	High	Load
AlarmRate		Number of alarms firing per second.	5.0 and above	High	Load

Parameters

The following parameters are valid for this counter category:

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Instance Name

Instance name to monitor. Select the Instance name that you want to monitor from the list of available instances. The default value is the first instance in the list.

You can specify one or more instances for monitoring. In any combination, select values from the discovered list, or enter values manually.

Primary Data Point

The datapoint type and the parameters specified in the task determine your datapoint. See WebSphere IDPs (Intelligent Data Points): Long.

Interval

Recommended minimum is 5 minutes.

WebSphere Bean Module Counters

The counters discovered for the WebSphere Bean category are determined by the level of metrics you set in WebSphere. The WebSphere Bean data counters may include the following counters:

Counter Name (6.0 and greater)	Counter Name (5.0 or earlier)	Description	WebSphere Version	Level of Metrics	Data Point Type
CreateCount	creates	Number of times beans were created.	3.5.5 and above	Low	Long

RemoveCount	removes	Number of times beans were removed.	3.5.5 and above	Low	Long
PassivateCount	passivates	Number of times beans were passivated (entity and stateful).	3.5.5 and above	Low	Long
ActivateCount	activates	Number of times beans were activated (entity and stateful).	3.5.5 and above	Low	Long
LoadCount	persistence loads	Number of times bean data was loaded from persistent storage (entity).	3.5.5 and above	Low	Long
StoreCount	persistence stores	Number of times bean data was stored in persistent storage (entity).	3.5.5 and above	Low	Long
InstantiateCount	instantiations	Number of times bean objects were instantiated.	3.5.5 and above	Low	Long
FreedCount	destroys	Number of times bean objects were freed.	3.5.5 and above	Low	Long
Ready Count	Num Ready Beans	Number of concurrently ready beans (entity and session). This counter was called concurrent active in Versions 3.5.5+ and 4.0.	3.5.5 and above	High	Load

LiveCount	concurrent live	Number of concurrently live beans.	3.5.5 and above	High	Load
MethodResponseTime	avg method rsp time	Average response time in milliseconds on the bean methods (home, remote, local).	3.5.5 and above	High	Long
CreateTime	avg method rsp time for create	Average time in milliseconds a bean create call takes, including the time for the load, if any.	5.0	Medium	Long
LoadTime	avg method rsp time for load	Average time in milliseconds for loading the bean data from persistent storage (entity).	5.0	Medium	Long
StoreTime	avg method rsp time for store	Average time in milliseconds for storing the bean data to persistent storage (entity).	5.0	Medium	Long
RemoveTime	avg method rsp time for remove	Average time in milliseconds a bean entry call takes, including the time at the database, if any.	5.0	Medium	Long
MethodCallCount	total method calls	Total number of method calls.	3.5.5 and above	High	Long

ActivationTime	avg method rsp time for activation	Average time in milliseconds a beanActivate call takes, including the time at the database, if any.	5.0	Medium	Long
PassivationTime	avg method rsp time for passivation	Average time in milliseconds a beanPassivate call takes, including the time at the database, if any.	5.0	Medium	Long
ActiveMethodCount	active methods	Number of concurrently active methods - number of methods called at the same time.	3.5.5 and above	High	Long
RetrieveFromPoolCount	Per method invocations	Number of calls to the bean methods (home, remote, local).	3.5.5 and above	Max	Long
RetrieveFromPoolSuccessCount	Per method rsp time	Average response time in milliseconds on the bean methods (home, remote, local).	3.5.5 and above	Max	Long
ReturnsToPoolCount	Per method concurrent invocations	Number of concurrent invocations to call a method.	5.0	Max	Load
RetrieveFromPoolCount	getsFromPool	Number of calls retrieving an object from the pool (entity and stateless).	3.5.5 and above	Low	Long

RetrieveFromPoolSuccessCount	getsFound	Number of times a retrieve found an object available in the pool (entity and stateless).	3.5.5 and above	Low	Long
ReturnsToPoolCount	returnsToPool	Number of calls returning an object to the pool (entity and stateless).	3.5.5 and above	Low	Long
ReturnsDiscardCount	returnsDiscarded	Number of times the returning object was discarded because the pool was full (entity and stateless).	3.5.5 and above	Low	Long
DrainsFromPoolCount	drainsFromPool	Number of times the daemon found the pool was idle and attempted to clean it (entity and stateless).	3.5.5 and above	Low	Long
DrainSize	avgDrainSize	Average number of objects discarded in each drain (entity and stateless).	3.5.5 and above	Medium	Long
PooledCount	avgPoolSize	Number of objects in the pool (entity and stateless).	3.5.5 and above	High	Load
MessageCount	messageCount	Number of messages delivered to the bean on Message method (message	5.0	Low	Long

		driven beans).			
MessageBackoutCount	messageBackoutCount	Number of messages failed to be delivered to the bean on Message method (message driven beans).	5.0	Low	Long
WaitTime	serverSessionWait	Average time to obtain a Server Session from the pool (message drive bean).	5.0	Medium	Long
ServerSessionPoolUsage	serverSessionUsage	Percentage of Server Session pool in use (message driven).	5.0	High	Load

Parameters

The following parameters are valid for this counter category:

Enterprise Beans (WebSphere Versions 3 and 4)

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Container

Name of bean container to monitor.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Bean

Name of Enterprise JavaBeans (EJB) to monitor.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Enterprise Beans (WebSphere Version 5)

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Application

Name of application to monitor.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Jar File

Name of jar file to monitor.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

EJB Type

Type of Enterprise JavaBeans (EJB) to monitor.

You can specify one or more types for monitoring. In any combination, select values from the discovered list, or enter values manually.

Bean

Name of Enterprise JavaBeans (EJB) to monitor.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Primary Data Point

The datapoint type and the parameters specified in the task determine your datapoint. See WebSphere IDPs (Intelligent Data Points): Long or Load.

Interval

Recommended minimum is 5 minutes.

Information presented in the table on this page is:
Reprinted Courtesy of International Business Machines Corporation copyright (2005) (c) International Business Machines Corporation.

WebSphere Cache Module Counters

The counters discovered for the Cache category are determined by the level of metrics you set in WebSphere. The Cache data counters may include the following counters:

Counter Name (6.0 and greater)	Counter Name (5.0 or earlier)	Description	WebSphere Version	Level of Metrics	Data Point Type
MaxInMemoryCacheEntryCount	maxInMemoryCacheSize	Maximum number of in-memory cache entries.	5.0 and above	Low	Long
InMemoryCacheEntryCount	inMemoryCacheSize	Current number of in-memory cache entries.	5.0 and above	Low	Long
TimeoutInvalidationCount	totalTimeoutInvalidation	Aggregate of template timeouts and disk timeouts.	5.0 and above	Low	Long
HitsInMemoryCount	hitsInMemory	Requests for this cacheable object served from memory.	5.0 and above	Low	Long
HitsOnDiskCount	hitsOnDisk	Requests for this cacheable object served from disk.	5.0 and above	Low	Long
ExplicitInvalidationCount	explicitInvalidations	Total explicit invalidation issued for this template.	5.0 and above	Low	Long
LruInvalidationCount	lruInvalidations	Cache entries evicted from memory by a Least Recently Used algorithm. These entries are passivated to disk if disk overflow is enabled.	5.0 and above	Low	Long
TimeoutInvalidationCount??????	timeoutInvalidations	Cache entries evicted from memory	5.0 and above	Low	Long

		and/or disk because their timeout has expired.			
InMemoryAndDiskCacheEntryCount	Entries	Current number of cache entries created from this template. Refers to the per-template equivalent of totalCacheSize.	5.0 and above	Low	Long
MissCount	Misses	Requests for this cacheable object that were not found in the cache.	5.0 and above	Low	Long
ClientRequestCount	RequestFromClient	Requests for this cacheable object generated by applications running on the application server.	5.0 and above	Low	Long
DistributedRequestCount	requestsFromJVM	Requests for this cacheable object generated by cooperating caches in this cluster.	5.0 and above	Low	Long
ExplicitMemoryInvalidationCount	explicitInvalidationsFromMemory	Explicit invalidations resulting in an entry being removed from memory.	5.0 and above	Low	Long
ExplicitDiskInvalidationCount	explicitInvalidationsFromDisk	Explicit invalidations resulting in an entry being removed from disk.	5.0 and above	Low	Long
ExplicitInvalidationCount	explicitInvalidationsNoOp	Explicit	5.0 and	Low	Lon

		invalidations received for this template where no corresponding entry exists.	above		g
LocalExplicitInvalidationCount	explicitInvalidationsLocal	Explicit invalidations generated locally, either programmatic ally or by a cache policy.	5.0 and above	Low	Lon g
RemoteExplicitInvalidationCou nt	explicitInvalidationsRemote	Explicit invalidations received from a cooperating JVM in this cluster.	5.0 and above	Low	Lon g
RemoteCreationCount	remoteCreations	Entries received from cooperating dynamic caches.	5.0 and above	Low	Lon g

Parameters

The following parameters are valid for this counter category:

Dynamic Cache (WebSphere Version 5)

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Instance Name

Instance name to monitor. Select the instance name that you want to monitor from the list of available instances. The default value is the first instance in the list.

You can specify one or more instances for monitoring. In any combination, select values from the discovered list, or enter values manually.

Primary Data Point

The datapoint type and the parameters specified in the task determine your datapoint. See WebSphere IDPs (Intelligent Data Points): Long.

Interval

Recommended minimum is 5 minutes.

WebSphere Connection Pool Module Counters

The counters discovered for the JDBC Connection Pool category are determined by the level of metrics you set in WebSphere. The JDBC Connection Pool data counters may include the following listed counters.

Performance Monitoring Infrastructure (PMI) collects performance data for 4.0 and 5.0 JDBC data sources. For a 4.0 data source, the data source name is used. For a 5.0 data source, the Java Naming and Directory Interface (JNDI) name is used.

Counter Name (6.0 and greater)	Counter Name (5.0 or earlier)	Description	WebSphere Version	Level of Metrics	Data Point Type
CreateCount	Creates	Total number of connections created.	3.5.5 and above	Low	Long
PoolSize	Avg Pool Size	Average pool size.	3.5.5 and above	High	Bounded Range Statistic
FreePoolSize	Free Pool Size	Average free pool size.	5.0	High	Bounded Range Statistic
AllocateCount	Allocates	Total number of connections allocated.	3.5.5 and above	Low	Long
ReturnCount	Returns	Total number of connections returned.	4.0 and above	Low	Long
WaitingThreadCount	Avg Waiting Threads	Number of threads that are currently waiting for a connection.	3.5.5 and above	High	Stat
FaultCount	Connection Pool Faults	Total number of faults, such as, timeouts, in connection pool.	3.5.5 and above	Low	Long
CloseCount	Destroys	Number of times bean objects were freed.	3.5.5 and above	Low	Long

WaitTime	Avg Wait Time	Average waiting time in milliseconds until a connection is granted.	5.0	Medium	Long
UseTime	Avg Time in Use	Average time a connection is used.	5.0	Medium	Long
PercentUsed	Percent Used	Average percent of the pool that is in use.	3.5.5 and above	High	Stat
PercentMaxed	Percent Maxed	Average percent of the time that all connections are in use	3.5.5 and above	High	Stat
PrepStmtCacheDiscardCount	Statement Cache discard count	Total number of statements discarded by the LRU algorithm of the statement cache.	4.0 and above	Low	Long
ManagedConnectionCount	Number Managed Connections	Number of Managed Connection objects in use.	5.0	Low	Long
ConnectionHandleCount	Number Connections	Current number of connection objects in use	5.0	Low	Long
JDBCTime	JDBC Operation Timer	Amount of time in milliseconds spent executing in the JDBC driver.	5.0	Medium	Long
	Concurrent Waiters				

Parameters

The following parameters are valid for this counter category:

JDBC Connection Pools (Versions 3 and 4)

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, enter values manually, or enter wildcard patterns.

Data Source

Name of data source.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

JDBC Connection Pools (Version 5)

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Provider

Name of data source provider to monitor.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Data Source

Name of data source to monitor.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Primary Data Point

The datapoint type and the parameters specified in the task determine your datapoint. See WebSphere IDPs (Intelligent Data Points): Long , Stat , or Bounded Range Statistic.

Interval

Recommended minimum is 5 minutes.

Information presented in the table on this page is:
Reprinted Courtesy of International Business Machines Corporation copyright (2005) (c) International Business Machines Corporation.

WebSphere DCS Stack Counters

The counters discovered for the WebSphere DCS Stack category are determined by the level of metrics you set in WebSphere. The WebSphere DCS Stack data counters may include the following counters:

Counter Name (6.0 and greater)	Counter Name (5.0 or earlier)	Description	WebSphere Version	Level of Metrics	Data Point Type
Number of message buffer reallocations		Number of message buffer reallocations due to inadequate buffer size. If this number is larger than 20 percent of the number of sent messages, you may want to contact IBM Support.	6.0 and above	Medium	Long
Outgoing message size		Minimal, maximal, and average size (in bytes) of the messages that were sent through the DCS stack.	6.0 and above	High	(AverageStatistic)
Number of sent messages		Number of messages sent through the DCS stack.	6.0 and above	High	Long
Incoming message size		Minimal, maximal and average size (in bytes) of the messages that were received by the DCS stack.	6.0 and above	High	(AverageStatistic)
Number of received messages		Number of messages received by the DCS stack.	6.0 and above	High	Long
Amount of time needed for the synchronization procedure to complete		Amount of time needed to guarantee that all view members are synchronized.	6.0 and above	High	Stat
Number of messages retransmitted by local member during the view change		Number of messages that were retransmitted during the view change to ensure synchronization with other members.	6.0 and above	High	(AverageStatistic)

Number of times that the synchronization procedure timed out		Number of times that the synchronization procedure timed out.	6.0 and above	Medium	Long
Number of times that a high severity congestion event for outgoing messages was raised		Number of times that a high severity congestion event for outgoing messages was raised.	6.0 and above	Medium	Long
Coalesce Time		Measures the amount of time it actually takes to coalesce a view.	6.0 and above	Medium	Stat
Join View Change Time		Measures the time to do a merge view change. The DCS stack is blocked during this time.	6.0 and above	High	Stat
Remove View Change Time		Measures the time to do a split view change. DCS stack is blocked during this time.	6.0 and above	High	Stat
Number of suspicions		Measures the number of times that the local member suspected other members.	6.0 and above	High	Long
Number of view changes		Number of times that this member underwent view changes.	6.0 and above	Medium	Long
View group size		Measures the size of the group the local member belongs to.	6.0 and above	Medium	(AverageStatistic)

Parameters

The following parameters are valid for this counter category:

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Instance Name

Instance name to monitor. Select the Instance name that you want to monitor from the list of available instances. The default value is the first instance in the list.

You can specify one or more instances for monitoring. In any combination, select values from the discovered list, or enter values manually.

Primary Data Point

The datapoint type and the parameters specified in the task determine your datapoint. See WebSphere IDPs (Intelligent Data Points): Long.

Interval

Recommended minimum is 5 minutes.

WebSphere High Availability Manager Counters

The counters discovered for the WebSphere High Availability Manager category are determined by the level of metrics you set in WebSphere. The WebSphere High Availability Manager data counters may include the following counters:

Counter Name (6.0 and greater)	Counter Name (5.0 or earlier)	Description	WebSphere Version	Level of Metrics	Data Point Type
Number of local groups		Total number of local groups.	6.0 and above	High	Load
Group state rebuild time		Time taken in milliseconds to rebuild the global group state. During the rebuild time, no fail-over can happen. If this time is too high and is unacceptable for the desired availability, you may want to increase the number of coordinators. For proper operation of this counter, you must host the active coordinator in an application server	6.0 and above	High	Stat

		other than the deployment manager.			
Number of bulletin-board subjects		Total number of subjects managed.	6.0 and above	High	Load
Number of bulletin-board subscriptions		Total number of bulletin-board subscriptions.	6.0 and above	High	Load
Bulletin-board rebuild time		Time taken in milliseconds to rebuild the global state of the bulletin-board. During this time no messages will be received by the subscribers. If this time is too high, and is unacceptable, you may want to increase the number of coordinators. For proper operation of this counter, you must host the active coordinator in an application server other than the deployment manager.	6.0 and above	High	Stat
Number of local bulletin-board subjects		Total number of subjects being posted to locally. The number includes the proxy postings (if any) done by the core group bridge service on behalf of servers belonging to different WebSphere cells.	6.0 and above	High	Load
Number of local bulletin-board subscriptions		Total number of local subject subscriptions. The number includes the proxy subscriptions (if any) done by the core group bridge service on behalf of servers belonging to different WebSphere cells.	6.0 and above	High	Stat

Parameters

The following parameters are valid for this counter category:

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Instance Name

Instance name to monitor. Select the Instance name that you want to monitor from the list of available instances. The default value is the first instance in the list.

You can specify one or more instances for monitoring. In any combination, select values from the discovered list, or enter values manually.

Primary Data Point

The datapoint type and the parameters specified in the task determine your datapoint. See WebSphere IDPs (Intelligent Data Points): Long.

Interval

Recommended minimum is 5 minutes.

WebSphere J2C Connection Pool Module Counters

The counters discovered for the J2C Connection Pool category are determined by the level of metrics you set in WebSphere. The J2C Connection Pool data counters may include the following counters:

Counter Name (6.0 and greater)	Counter Name (5.0 or earlier)	Description	WebSphere Version	Level of Metrics	Data Point Type
ManagedConnectionCount	Number managed connections	Number of Managed Connection objects in use.	5.0 and above	Low	Long
ConnectionHandleCount	Number connections	Current number of connection objects in use.	5.0 and above	Low	Long
CreateCount	Number managed connections created	Total number of connections created.	5.0 and above	Low	Long
CloseCount	Number managed connections destroyed	Total number of connections destroyed.	5.0 and above	Low	Long

AllocateCount	Number managed connections allocated	Total number of connections allocated.	5.0 and above	Low	Long
FreedCount	Number managed connections freed	Total number of connections freed.	5.0 and above	Low	Long
FaultCount	faults	Number of faults, such as timeouts, in the connection pool.	5.0 and above	Low	Long
FreePoolSize	free pool size	Number of free connections in the pool.	5.0 and above	High	Stat
PoolSize	pool size	Pool size.	5.0 and above	High	Stat
WaitingThreadCount	concurrent waiters	Average number of threads concurrently waiting for a connection.	5.0 and above	High	Load
PercentUsed	Percent used	Average percent of the pool that is in use.	5.0 and above	High	Load
PercentMaxed	Percent maxed	Average percent of the time that all connections are in use.	5.0 and above	High	Load
WaitTime	Average wait time	Average waiting time in milliseconds until a connection is granted.	5.0 and above	Medium	Long
UseTime	Average use time	Average time in milliseconds that connections are in use.	5.0 and above	Medium	Long

Parameters

The following parameters are valid for this counter category:

J2C Connection Pools (WebSphere Version 5)

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Instance Name

Instance name to monitor. Select the instance name that you want to monitor from the list of available instances. The default value is the first instance in the list.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Primary Data Point

The datapoint type and the parameters specified in the task determine your datapoint. See WebSphere IDPs (Intelligent Data Points): Long, Load, or Stat.

Interval

Recommended minimum is 5 minutes.

WebSphere Java Virtual Machine (JVM) Runtime Module Counters

The counters discovered for the Java Virtual Machine (JVM) category are determined by the level of metrics you set in WebSphere. The JVM data counters may include the following counters:

Counter Name (6.0 and greater)	Counter Name (5.0 or earlier)	Description	WebSphere Version	Level of Metrics	Data Point Type
FreeMemory	Free memory	Free memory in JVM run time.	3.5.5 and above	Low	Long
UsedMemory	Used memory	Used memory in JVM run time.	3.5.5 and above	Low	Long
HeapSize	Total memory	Total memory in JVM run time.	3.5.5 and above	High	Long
UpTime	Up time	The amount of time the JVM is running.	5.0 and above	Low	Long
GCCount	Number garbage collection calls	Number of garbage collection calls. This counter is not available unless `-XrunpmiJvmpiProfiler` is set when starting the JVM.	4.0 and above	Max	Long
GCIntervalTime	Average time between garbage collection	Average garbage collection in seconds between two garbage collection. This counter is not available unless -`XrunpmiJvmpiProfiler` is set when starting the JVM.	4.0 and above	Max	Long

GCTime	Average garbage collection duration	Average duration of a garbage collection. This counter is not available unless -`XrunpmiJvmpiProfiler` is set when starting the JVM.	4.0 and above	Max	Long
WaitsForLockCount	num waits for a lock	Number of times that a thread waits for a lock.This counter is not available unless -`XrunpmiJvmpiProfiler` is set when starting the JVM.	4.0 and above	Max	Long
WaitForLockTime	avg time waiting for lock	Average time that a thread waits for a lock. This counter is not available unless -`XrunpmiJvmpiProfiler` is set when starting the JVM.	4.0 and above	Max	Long
ObjectAllocateCount	Number of objects allocated	Number of objects allocated in heap. This counter is not available unless -`XrunpmiJvmpiProfiler` is set when starting the JVM.	4.0 and above	Max	Long
ObjectMovedCount					
	Number of objects found	Number of objects in heap. This counter is not available unless -`XrunpmiJvmpiProfiler` is set when starting the JVM.	4.0 and above	Max	Long
ObjectFreedCount	Number of objects freed	Number of objects freed in heap. This counter is not available unless -`XrunpmiJvmpiProfiler` is set when starting the JVM.	4.0 and above	Max	Long
ThreadStartedCount		Number of threads started. This counter is not available unless the -XrunpmiJvmpiProfiler option is set when starting the JVM.	4.0 and above		
ThreadEndedCount		Number of failed threads. This counter is not available unless the -XrunpmiJvmpiProfiler option is set when starting the JVM.	4.0 and above		

Parameters

The following parameters are valid for this counter category:

JVM Runtime (WebSphere All Versions)

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Primary Data Point

The datapoint type and the parameters specified in the task determine your datapoint. See WebSphere IDPs (Intelligent Data Points): Long.

Interval

Recommended minimum is 5 minutes.

Information presented in the table on this page is:
Reprinted Courtesy of International Business Machines Corporation copyright (2005) (c) International Business Machines Corporation.

WebSphere Object Pool Counters

The counters discovered for the WebSphere Object Pool category are determined by the level of metrics you set in WebSphere. The WebSphere Object Pool data counters may include the following counters:

Counter Name (6.0 and greater)	Counter Name (5.0 or earlier)	Description	WebSphere Version	Level of Metrics	Data Point Type
ObjectsCreatedCoun		Total number of objects created.	5.0 and above	High	Long
ObjectsAllocatedCount		Number of objects requested from the pool.	5.0 and above	High	Long
ObjectsReturnedCount		Number of objects returned to the pool.	5.0 and above	High	Long
IdleObjectsSize		Average number of idle object instances in the pool.	5.0 and above	High	Load

Parameters

The following parameters are valid for this counter category:

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Instance Name

Instance name to monitor. Select the Instance name that you want to monitor from the list of available instances. The default value is the first instance in the list.

You can specify one or more instances for monitoring. In any combination, select values from the discovered list, or enter values manually.

Primary Data Point

The datapoint type and the parameters specified in the task determine your datapoint. See WebSphere IDPs (Intelligent Data Points): Long.

Interval

Recommended minimum is 5 minutes.

WebSphere ORB Perf Module Counters

The counters discovered for the Object Request Broker (ORB) category are determined by the level of metrics you set in WebSphere. The ORB data counters may include the following counters:

Counter Name (6.0 and greater)	Counter Name (5.0 or earlier)	Description	WebSphere Version	Level of Metrics	Data Point Type
LookupTime	referenceLookupTime	The time (in milliseconds) to look up an object reference before method dispatch can be carried out.	5.0 and above	Medium	Long
RequestCount	numRequest	The total number of requests sent to the ORB.	5.0 and above	Low	Long
ConcurrentRequestCount	concurrentRequests	The number of requests that are concurrently processed by the ORB.	5.0 and above	High	Load
ProcessingTime	processingTime	The time (in milliseconds) it takes a registered portable interceptor to run.	5.0 and above	Medium	Long

Parameters

The following parameters are valid for this counter category:

Object Request Broker (WebSphere Version 5)

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Instance Name

Instance name to monitor. Select the instance name that you want to monitor from the list of available instances. The default value is the first instance in the list.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Primary Data Point

The datapoint type and the parameters specified in the task determine your datapoint. See WebSphere IDPs (Intelligent Data Points): Long or Load.

Interval

Recommended minimum is 5 minutes.

Information presented in the table on this page is:
Reprinted Courtesy of International Business Machines Corporation copyright (2005) (c) International Business Machines Corporation.

WebSphere Scheduler Module Counters

The counters discovered for the WebSphere Scheduler category are determined by the level of metrics you set in WebSphere. The WebSphere Scheduler data counters may include the following counters:

Counter Name (6.0 and greater)	Counter Name (5.0 or earlier)	Description	WebSphere Version	Level of Metrics	Data Point Type
TaskFailureCount		Number of tasks that failed to run.	5.0 and above	High	Long
TaskFinishCount		Number of tasks that ran successfully.	5.0 and above	High	Long
PollCount		Number of poll cycles completed for all daemon threads.	5.0 and above	High	Long
TaskFinishRate		Number of tasks run per second.	5.0 and above	High	Load

TaskCollisionRate		Number of collisions encountered per second between competing poll daemons.	5.0 and above	High	Load
PollQueryDuration		Start time in milliseconds for each poll daemon thread's database poll query.	5.0 and above	High	Load
RunDuration		Time in milliseconds taken to run a task..	5.0 and above	High	Load
TaskExpirationRate		Number of tasks in a poll query.	5.0 and above	High	Load
TaskDelayDuration		Period of time in seconds that the task is delayed.	5.0 and above	High	Load
PollDuration		Number of seconds between poll cycles.	5.0 and above	High	Load
TaskRunRate		Number of tasks run by each poll daemon thread. (Multiply this by the number of poll daemon threads to get the tasks run per effective poll cycle.)	5.0 and above	High	Load

Parameters

The following parameters are valid for this counter category:

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Instance Name

Instance name to monitor. Select the instance name that you want to monitor from the list of available instances. The default value is the first instance in the list.

You can specify one or more instances for monitoring. In any combination, select values from the discovered list, or enter values manually.

Primary Data Point

The datapoint type and the parameters specified in the task determine your datapoint. See WebSphere IDPs (Intelligent Data Points): Long.

Interval

Recommended minimum is 5 minutes.

WebSphere Servlet Sessions Module Counters

The counters discovered for the Servlet Sessions category are determined by the level of metrics you set in WebSphere. The Servlet Sessions data counters may include the following counters:

Counter Name (6.0 and greater)	Counter Name (5.0 or earlier)	Description	WebSphere Version	Level of Metrics	Data Point Type
CreateCount	createdSessions	Number of sessions created.	3.5.5 and above	Low	Long
InvalidateCount	invalidatedSessions	Number of sessions invalidated.	3.5.5 and above	Low	Long
LifeTime	sessionLifeTime	Average session lifetime.	3.5.5 and above	Medium	Long
ActiveCount	activeSessions	Number of concurrently active sessions. A session is active if WebSphere is currently processing a request that uses that session.	3.5.5 and above	High	Load
LiveCount	liveSession	Number of sessions that are currently cached in memory.	5.0 and above	High	Load
NoRoomForNewSessionCount	NoRoomForNewSession	Applies only to session in memory with AllowOverflow=false. The number of times that a request for a new session cannot be handled because it would exceed the	5.0	Low	Long

		maximum session count.			
CacheDiscardCount	cacheDiscards	Number of session objects that have been forced out of the cache. (An LRU algorithm removes old entries to make room for new sessions and cache misses). Applicable only for persistent sessions.	5.0	Low	Long
ExternalReadTime	externalReadTime	Time (in milliseconds) taken in reading the session data from persistent store. For multi-row sessions, the metrics are for the attribute; for single-row sessions, the metrics are for the whole session. Applicable only for persistent sessions. When using a JMS persistent store, you have the choice of whether to serialize the data being replicated. If you choose not to serialize the data, the counter is not available.	5.0 4	Mediu m	Long
ExternalReadSize	externalReadSize	Size of session data read from persistent store. Applicable only for (serialized) persistent sessions; similar to externalReadTime above.	5.0	Mediu m	Lon g
ExternalWriteTime	externalWriteTime	Time (milliseconds) taken to write the session data from the persistent store. Applicable only for (serialized) persistent sessions. Similar to	5.0	Mediu m	Lon g

		externalReadTime described above.			
ExternalWriteSize	externalWriteSize	Size of session data written to persistent store. Applicable only for (serialized) persistent sessions. Similar to externalReadTime described above.	5.0	Mediu m	Lon g
AffinityBreakCount	affinityBreaks	The number of requests received for sessions that were last accessed from another Web application. This can indicate failover processing or a corrupt plug-in configuration.	5.0	Low	Long
SessionObjectSize	serializableSessObjSize	The size in bytes of (the attributes that can be serialized) in-memory sessions. Only count session objects that contain at least one attribute object that can be serialized. Note that a session may contain some attributes that can be serialized and some that are not. The size in bytes is at a session level.	5.0	Max	Long
TimeSinceLastActivated	timeSinceLastActivate d	The time difference in milliseconds between previous and current access time stamps. Does not include session time out.	5.0	Mediu m	Long
TimeoutInvalidationCount	invalidatedViaTimeou t	The number of requests for a session that no CountStatistic exists, presumably because the session timed out.	5.0	Low	Long

ActivateNonExistSessionCount	attemptToActivateNotExistentSession	Number of requests for a session that no longer exists, presumably because the session timed out. Use this counter to help determine if the timeout is too short.	5.0	Low	Long

Parameters

The following parameters are valid for this counter category:

Servlet Sessions (WebSphere Versions 3 and 4)

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Servlet Sessions (WebSphere Version 5)

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Application

Name of application to monitor.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

War File

Name of war file to monitor.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Primary Data Point

The datapoint type and the parameters specified in the task determine your datapoint. See WebSphere IDPs (Intelligent Data Points): Long or Load.

Interval

Recommended minimum is 5 minutes.

Information presented in the table on this page is:
Reprinted Courtesy of International Business Machines Corporation copyright (2005) (c) International Business Machines Corporation.

WebSphere System Module Counters

The counters discovered for the System category are determined by the level of metrics you set in WebSphere. The System data counters may include the following counters:

Counter Name (6.0 and greater)	Counter Name (5.0 or earlier)	Description	WebSphere Version	Level of Metrics	Data Point Type
CPUUsageSinceLastMeasurement	percentCpuUsage	Average system CPU utilization taken over the time interval since the last reading. Because the first call is required to perform initialization, an invalid value such as 0 is returned. All subsequent calls return the expected value. On SMP machines, the value returned is the utilization averaged over all CPUs.	5.0	Low	Long
FreeMemory	freeMemory	The amount of real free memory available on the system. Real memory that is not allocated is only a lower bound on available real memory, since	5.0	Low	Long

		many operating systems take some of the otherwise unallocated memory and use it for additional I/O buffering. The exact amount of buffer memory that can be freed up is dependent on both the platform and the application(s) running on it.			
CPUUsageSinceServerStarted	avgCpuUtilization	The average percentCpuUsage that is busy after the server is started.	5.0	Medium	Long

Parameters

The following parameters are valid for this counter category:

System Performance (WebSphere Version 5)

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Primary Data Point

The datapoint type and the parameters specified in the task determine your datapoint. See WebSphere IDPs (Intelligent Data Points): Long.

Interval

Recommended minimum is 5 minutes.

WebSphere Thread Pool Module Counters

The counters discovered for the Thread Pool category are determined by the level of metrics you set in WebSphere. The Thread Pool data counters may include the following counters:

Counter Name (6.0 and greater)	Counter Name (5.0 or earlier)	Description	WebSphere Version	Level of Metrics	Data Point Type
CreateCount	Thread creates	Total number of threads created.	3.5.5 and above	Low	Long
DestroyCount	Thread destroys	Total number of threads destroyed.	3.5.5 and above	Low	Long
ActiveCount	Active threads	Number of concurrently active threads.	3.5.5 and above	High	Load
PoolSize	Pool size	Average number of threads in pool.	3.5.5 and above	High	Load
PercentMaxed	Percent maxed	Average percent of the time that all threads are in use.	3.5.5 and above	High	Load

Parameters

The following parameters are valid for this counter category:

Thread Pools (WebSphere All Versions)

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Pool

Name of thread pool to monitor.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Primary Data Point

The datapoint type and the parameters specified in the task determine your datapoint. See WebSphere IDPs (Intelligent Data Points): Long or Load.

Interval

Recommended minimum is 5 minutes.

Information presented in the table on this page is:
Reprinted Courtesy of International Business Machines Corporation copyright (2005) (c) International Business Machines Corporation.

WebSphere Transaction Module Counters

The counters discovered for the Transaction category are determined by the level of metrics you set in WebSphere. The Transaction data counters may include the following counters:

Counter Name (6.0 and greater)	Counter Name (5.0 or earlier)	Description	WebSphere Version	Level of Metrics	Data Point Type
GlobalBegunCount	Number global transactions begun	Total number of global transactions begun on server.	4.0 and above	Low	Long
GlobalInvolvedCount	Number global transactions involved	Total number of global transactions involved on server (for example, begun and imported).	4.0 and above	Low	Long
LocalBegunCount	Number local transactions begun	Total number of local transactions begun on server.	4.0 and above	Low	Long
ActiveCount	Active global transactions	Number of concurrently active global transactions.	3.5.5 and above	Low	Load
LocalActiveCount	Active local transactions	Number of concurrently active local transactions.	4.0 and above	Low	Load
GlobalTranTime	Global transactions duration	Average duration of global transactions.	3.5.5 and above	Medium	Stat
LocalTranTime	Local transaction duration	Average duration of local transactions.	4.0 and above	Medium	Stat
GlobalBeforeCompletionTime	Local transactions before_completion time	Average duration of before_completion for local transactions.	4.0 and above	Medium	Stat
GlobalCommitTime	Global transaction commit time	Average duration of commit for global transactions.	4.0 and above	Medium	Stat
GlobalPrepareTime	Global transaction prepare time	Average duration of prepare for global	4.0 and above	Medium	Stat

		transactions.			
LocalBeforeCompletionTime	Local transaction before_completion time	Average duration of before_completion for local transactions.	4.0 and above	Medium	Stat
LocalCommitTime	Local transaction commit time	Average duration of commit for local transactions.	4.0 and above	Medium	Stat
CommittedCount	Number global transactions committed	Total number of global transactions committed.	3.5.5 and above	Low	Long
RolledbackCount	Number of global transactions rolled back	Total number of global transactions rolled back.	3.5.5 and above	Low	Long
OptimizationCount	Number global transactions optimized	Number of global transactions converted to single phase for optimization.	4.0 and above	Low	Long
LocalCommittedCount	Number of local transactions committed	Number of local transactions committed.	4.0 and above	Low	Long
LocalRolledbackCount	Number of local transactions rolled back	Number of local transactions rolled back.	4.0 and above	Low	Long
GlobalTimeoutCount	Number of global transactions timed out	Number of global transactions timed out.	4.0 and above	Low	Long
LocalTimeoutCount	Number of local transactions timed out	Number of local transactions timed out.	4.0 and above	Low	Long

Parameters

The following parameters are valid for this counter category:

Transactions (All Versions)

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Primary Data Point

The datapoint type and the parameters specified in the task determine your datapoint. See WebSphere IDPs (Intelligent Data Points): Long, Load, or Stat.

Interval

Recommended minimum is 5 minutes.

Information presented in the table on this page is:
Reprinted Courtesy of International Business Machines Corporation copyright (2005) (c) International Business Machines Corporation.

WebSphere Web App Module Counters

The counters discovered for the Web Application category are determined by the level of metrics you set in WebSphere. The Web Application data counters may include the following counters:

Counter Name (6.0 and greater)	Counter Name (5.0 or earlier)	Description	WebSphere Version	Level of Metrics	Data Point Type
LoadedServletCount	numLoadedServlets	Number of servlets that were loaded.	3.5.5 and above	Low	Long
ReloadCount	numReloads	Number of servlets that were reloaded.	3.5.5 and above	Low	Load
RequestCount	totalRequests	Total number of requests a servlet processed.	3.5.5 and above	Low	Long
ConcurrentRequests	concurrentRequests	Number of requests that are concurrently processed.	3.5.5 and above	High	Stat
ServiceTime	responseTime	Response time, in milliseconds, of a servlet request.	3.5.5 and above	Medium	Long
ErrorCount	numErrors	Total number of errors in a servlet or Java Server Page (JSP).	3.5.5 and above	Low	Long

Parameters

The following parameters are valid for this counter category:

Web Applications (Versions 3 and 4)

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Application

Name of application to monitor.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Servlet

Name of servlet to monitor.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Web Applications (Version 5)

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Application

Name of application to monitor.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

War File

Name of war file to monitor.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Servlet

Name of servlet to monitor.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Primary Data Point

The datapoint type and the parameters specified in the task determine your datapoint. See WebSphere IDPs (Intelligent Data Points): Long, Load, or Stat.

Interval

Recommended minimum is 5 minutes.

WebSphere Web Services Module Counters

The counters discovered for the WebSphere Web Service category are determined by the level of metrics you set in WebSphere. The WebSphere Web Service data counters may include the following counters:

Counter Name (6.0 and greater)	Counter Name (5.0 or earlier)	Description	WebSphere Version	Level of Metrics	Data Point Type
LoadedWebServiceCount		Number of loaded Web services.	5.02 and above	Low	Long
ReceivedRequestCount		Number of requests the service received.	5.02 and above	Low	Long
DispatchedRequestCount		Number of requests the service dispatched.	5.02 and above	Low	Long
ProcessedRequestCount		Number of requests the service successfully processed.	5.02 and above	Low	Stat
ResponseTime		Average response time, in milliseconds, for a successful request.	5.02 and above	High	Stat
RequestResponseTime		Average response time, in milliseconds, to prepare a request for dispatch.	5.02 and above	Medium	Stat
DispatchResponseTime		Average response time, in milliseconds, to dispatch a request.	5.02 and above	Medium	Stat
ReplyResponseTime		Average response time, in milliseconds, to prepare a reply after dispatch.	5.02 and above	Medium	Stat
PayloadSize		Average payload size in bytes of a received	5.02 and above	Medium	Stat

		request or reply.			
RequestPayloadSize		Average payload size in bytes of a request.	5.02 and above	Medium	Stat
ReplyPayloadSize		Average payload size in bytes of a reply.	5.02 and above	Medium	Stat

Parameters

The following parameters are valid for this counter category:

Node Name

Node or machine name to monitor. Select the node that you want to monitor from the list of available nodes. The default value is the first node in the list of available nodes.

You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

Server Name

Application server to monitor. Select the server name that you want to monitor from the list of available servers. The default value is the first application server in the list.

You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

Instance Name

Instance name to monitor. Select the instance name that you want to monitor from the list of available instances. The default value is the first instance in the list.

You can specify one or more instances for monitoring. In any combination, select values from the discovered list, or enter values manually.

Primary Data Point

The datapoint type and the parameters specified in the task determine your datapoint. See WebSphere IDPs (Intelligent Data Points): Long.

Interval

Recommended minimum is 5 minutes.

Information presented in the table on this page is:
Reprinted Courtesy of International Business Machines Corporation copyright (2005) (c) International Business Machines Corporation.

WebSphere MQ Counters

WebSphere MQ Remote Extended Counters

The following extended WebSphere MQ remote counters are provided in QALoad. These counters extend the monitoring of your WebSphere MQ system:

Channel Events	Queue Manager Connections

Channel Status	Queue Manager Events
Error Log Entries	Queue Manager Statistics
Percent Queue Depth	Queue Manager Up/Down
Performance Events	Queue Statistics
Queue Depth	

WebSphere MQ Channel Events

This counter reports the number of channel events for the current interval.

Parameters

Queue Manager

Queue manager you are monitoring.

Event Name

Specify the name(s) of the event(s) you want to monitor. All events on the queue are monitored unless event name(s) are selected.

Event Name	Description
Channel Activated	This condition is detected when a channel that has been waiting to become active,and for which a Channel Not Activated event has been generated, is now able to become active, because an active slot has been released by another channel.
Channel Auto-Definition Error	Automatic channel definition failed.
Channel Auto-Definition OK	Automatic channel definition succeeded.
Channel Conversion Error	This condition is detected when a channel is unable to carry out data conversion.
Channel Not Activated	The channel is unable to establish the connection because the limit on the number of active channels has been reached.
Channel Started	An instance of a channel has been successfully established
Channel Stopped	The channel was stopped.
Channel Stopped By User	The channel has been stopped by the operator.

Primary Data Point

The primary data point is the number of channel events for the specified queue for the current interval.

Intelligent Data Point

The intelligent data point displays the number of channel events, description of individual events (event name, date & time that the message was put on the event queue, name of the queue manager that put the message, queue associated with the event, and reason code).

Channel events are reported by channels as a result of conditions detected during their operation. For example, when a channel instance is stopped. Channel events are generated:

- By a command to start or stop a channel
- When a channel instance starts or stops
- When a channel receives a conversion error warning when getting a message.
- When an attempt is made to create a channel automatically; the event is generated whether the attempt succeeds or fails.

Interval

Recommended minimum is 5 minutes.

WebSphere MQ Channel Status

This counter reports the running state of a channel. This counter cannot be used for client-connection channels.

Parameters

Queue Manager

Queue manager you are monitoring.

Channel

Channel you are monitoring.

Primary Data Point

The primary data point is the running state of a channel.

- "1" if the channel is active.
- "0" if the channel is not active.
- "-1" if an error occurred.

Intelligent Data Point

The intelligent data point lists the queue manager, channel name, and status, or if an error occurred.

Interval

Recommended minimum is 5 minutes.

WebSphere MQ Error Log Entries

This counter reports the number of errors in the MQ error log file for the current interval. It uses standard Java file processing API functions to gather the information.

Note: This counter does not appear in the discovery data if the MQ instance was configured as remote in the agent manager.

Parameters

Error Number

Specify All Errors, a single error number, or an error number range to monitor.

Error Number Range	Description
AMQ3500-AMQ3999	WebSphere MQ for Windows messages.
AMQ4000-AMQ4999	WebSphere MQ for Windows NT User Interface messages.
AMQ5000-AMQ5999	Installable services messages.
AMQ6000-AMQ6999	Common services messages.
AMQ7000-AMQ7999	WebSphere MQ product messages.
AMQ8000-AMQ8999	WebSphere MQ administration messages.
AMQ9000-AMQ9999	Remote messages.

Primary Data Point

The primary data point is the number of errors in the error log file for the current interval.

Intelligent Data Point

There are three alternatives for what is returned for the intelligent data point. It depends on what you selected for the Error Number parameter.

The data point detail lists each error and the number of times it occurred within the interval.

One error chosen:

! Number of errors in interval that match the error number.

! Each error range and count for the range.

! Total errors during the interval.

Error range chosen:

! Number of errors in interval that are within chosen range.

! Top 10 errors in range.

! Error range and count for the range.

! Total errors during the interval.

All errors chosen:

! Number of errors in interval.

! Top 10 errors.

! Error and count.

! Error range and count for the range.

! Total errors during the interval.

Interval

Recommended minimum is 5 minutes.

WebSphere MQ Percent Queue Depth

This counter reports the current queue depth as a percentage of the defined maximum.

Parameters

Queue Manager

Queue manager you are monitoring.

Queue

Name of the queue you are monitoring.

Primary Data Point

The primary data point is the current queue depth as a percentage of the defined maximum.

Intelligent Data Point

The intelligent data point lists the queue manager, queue, current queue depth, and percent queue depth.

Interval

Recommended minimum is 5 minutes.

WebSphere MQ Performance Events

This counter reports the number of performance events for the current interval.

Parameters

Queue Manager

Queue manager you are monitoring.

Performance Event Queue

Name of the performance event queue that you are monitoring. The default value is SYSTEM.ADMIN.PERFM.EVENT.

Event Name

Specify the name(s) of the event(s) you want to monitor. All events on the queue are monitored unless event name(s) are selected.

Event Name	Description
Queue Depth High	Queue depth high limit reached or exceeded.
Queue Depth Low	Queue depth low limit reached or exceeded.
Queue Full	Queue already contains maximum number of messages.
Queue Service Interval High	No successful gets or puts have been detected within an interval greater than the limit specified in the Q Service Interval attribute.

Queue Service Interval OK	A successful get has been detected within an interval less than or equal to the limit specified in the Q Service Interval attribute.

Primary Data Point

The primary data point is the number of performance events for the specified queue during the current interval.

Intelligent Data Point

The intelligent data point displays the number of performance events, description of individual events (performance event type, event name, date & time that the message was put on the event queue, name of the queue manager that put the message, queue associated with the event, time since reset, high queue depth, message enqueue count, message dequeue count and reason code).

Performance events are notifications that a threshold condition has been reached by a resource. The conditions can affect the performance of applications that use a specified queue. Performance event types are:

- Queue Depth High
- Queue Depth Low
- Queue Full
- Queue Service Interval High
- Queue Service Interval OK

Performance event statistics are reset when a performance event occurs or a queue manager stops and restarts.

Interval

Recommended minimum is 5 minutes.

WebSphere MQ Queue Depth

This counter monitors the current depth of the specified queue.

Parameters

Queue Manager

Queue manager you are monitoring.

Queue

Name of the queue you are monitoring.

Primary Data Point

The primary data point is the number of messages on queue.

Intelligent Data Point

The intelligent data point lists the queue manager, queue, and queue depth.

Interval

Recommended minimum is 5 minutes.

WebSphere MQ Queue Manager Connections

This counter reports the current number of connections to a queue manager.

Parameters

Queue Manager

Queue manager you are monitoring.

Primary Data Point

The primary data point is the positive integer representing the number of connections or "-1" if an error occurred.

Intelligent Data Point

The intelligent data point lists the queue manager, the number of active connections, and the connection names. If an error occurred, then it displays the error number and description.

Interval

Recommended minimum is 5 minutes.

WebSphere MQ Queue Manager Events

This counter reports the number of queue manager events for the current interval.

Parameters

Queue Manager

Queue manager you are monitoring.

Queue Manager Event Queue

Name of the queue manager event queue you are monitoring. The default value is SYSTEM.ADMIN.QMGR.EVENT.

Event Name

Specify the name(s) of the event(s) to monitor. All events on the queue are monitored unless event name(s) are selected.

Event Name	Description
Alias Base Queue Type Error	The Base Q Name in the alias queue definition resolves to a queue that is not a local queue, or local definition of a remote queue.
Default Transmission Queue Type Error	Either a local definition of the remote queue was specified, or a queue-manager alias was being resolved, but in either case the XmitQName attribute in the local definition is blank.
Default Transmission Queue Usage Error	The queue defined by the DefXmitQNamequeue-manager attribute does not have a Usage attribute of MQUS_TRANSMISSION.
Get Inhibited	Gets inhibited for the queue.

Not Authorized	The user is not authorized for access.
Put Inhibited	Put calls inhibited for the queue.
Queue Manager Active	Queue manager created.
Queue Manager Not Active	Queue manager unavailable.
Queue Type Error	Queue type not valid.
Remote Queue Name Error	Remote queue name not valid.
Transmission Queue Type Error	Transmission queue not local.
Transmission Queue Usage Error	Transmission queue with wrong usage.
Unknown Alias Base Queue	The BaseQName in the alias queue attributes is not recognized as a queue name.
Unknown Default Transmission Queue	The XmitQName attribute in the local definition is blank.
Unknown Object Name	The Object Name in the object descriptor is not recognized for the specified object type.
Unknown Remote Queue Manager	An error occurred with the queue-name resolution.
Unknown Transmission Queue	The XmitQName attribute of the definition is not blank and not the name of a locally-defined queue.

Primary Data Point

The primary data point is the number of queue manager events for the current interval.

Intelligent Data Point

The intelligent data point displays the number of queue manager events, description of individual events (queue manager event type, event name, date & time that the message was put on the event queue, name of the queue manager that put the message, and reason code).

Queue manager events are events that are related to the definitions of resources within queue managers. For example, an application attempts to put a message to a queue that does not exist. Queue manager event types are: authority, inhibit, local, remote, and start/stop.

Event Type	Reason Code
Authority Events	! Not Authorized (type 1) ! Not Authorized (type 2) ! Not Authorized (type 3) ! Not Authorized (type 4)
Inhibit Events	! Get Inhibited

	! Put Inhibited
Local Events	! Alias Base Queue Type Error ! Unknown Alias Base Queue ! Unknown Object Name
Remote Events	! Default Transmission Queue Type Error ! Default Transmission Queue Usage Error ! Queue Type Error ! Remote Queue Name Error ! Transmission Queue Type Error ! Transmission Queue Usage Error ! Unknown Default Transmission Queue ! Unknown Remote Queue Manager ! Unknown Transmission Queue
Start and Stop Events	! Queue Manager Active ! Queue Manager Not Active

Interval

Recommended minimum is 5 minutes.

WebSphere MQ Queue Manager Statistics

This counter reports statistics describing a queue manager.

Parameters

Queue Manager

Queue manager you are monitoring.

Statistic

Specify the statistic to use as the primary data point.:

Authority Events	Reports the on/off value of Authority events. Authority events indicate that an authorization violation has been detected.
Automatic Channel Definition Events	Reports the on/off value of Automatic Channel Definition events. Automatic channel definition events indicate whether an automatic definition of a channel fails or succeeds.
Inhibit Events	Reports the on/off value of Inhibit events. Inhibit events indicate that an MQPUT or MQGET operation has been attempted against a queue, where the queue is inhibited for puts or gets respectively.
Local Events	Reports the on/off value of Local events. Local events indicate that an application (or the queue manager) has not been able to access a local queue, or other local object.

Performance Events	Reports the on/off value of Performance events. Performance events are notifications that a threshold condition has been reached by a resource.
Remote Events	Reports the on/off value of Remote events. Remote events indicate that an application (or the queue manager) cannot access a (remote) queue on another queue manager.
Start Stop Events	Reports the on/off value of these events. Start and stop events indicate that a queue manager has been started or has been requested to stop or quiesce.

Primary Data Point

The primary data point is the one of the following statistics as specified by the Statistics parameter:

- Authority Events
- Automatic Channel Definition Events
- Inhibit Events
- Local Events
- Performance Events
- Authority Events
- Automatic Channel Definition Events

Intelligent Data Point

The intelligent data point lists the queue manager and queue manager statistics. This counter reports the current state of the statistics, it does not report the statistics values as they progress through time. For dynamic information, monitor with the Queue Manager Event counter. As appropriate, any of the following information may be included:

Data	Description
Authority Events = <integer>	Variable that stores the on/off value of these events. Authority events indicate that an authorization violation has been detected.
Automatic Channel Definition Events = <integer>	Variable that stores the on/off value of these events. Automatic channel definition events indicate whether an automatic definition of a channel fails or succeeds.
Inhibit Events = <integer>	Variable that stores the on/off value of these events. Inhibit events indicate that an MQPUT or MQGET operation has been attempted against a queue, where the queue is inhibited for puts or gets respectively.
Local Events = <integer>	Variable that stores the on/off value of these events. Local events indicate that an application (or the queue manager) has not been able to access a local queue, or other local object.

Performance Events = <integer>	Variable that stores the on/off value of these events. Performance events are notifications that a threshold condition has been reached by a resource.
Remote Events = <integer>	Variable that stores the on/off value of these events. Remote events indicate that an application (or the queue manager) cannot access a (remote) queue on another queue manager.
Start Stop Events = <integer>	Variable that stores the on/off value of these events. Start and stop events indicate that a queue manager has been started or has been requested to stop or quiesce.
Cluster Workload Data = <wstring>	Cluster workload exit data.
Command Level = <integer>	Level of system control commands supported by the queue manager.

Interval

Recommended minimum is 5 minutes.

WebSphere MQ Queue Manager Up/Down

This counter monitors the running state of a queue manager.

Parameters

Queue Manager

Queue manager you are monitoring.

Primary Data Point

The primary data point is the running state of queue manager.

! "1" if the queue manager is running.

! "0" if the queue manager is not running.

! "-1" if an error occurred.

Intelligent Data Point

The intelligent data point lists the queue manager and whether the queue manager is up, down, or an error occurred.

Interval

Recommended minimum is 5 minutes.

WebSphere MQ Queue Statistics

This counter reports statistics describing a queue.

Parameters

Queue Manager

Queue manager you are monitoring.

Queue

Name of the queue you are monitoring.

Statistic

Specify the statistic to use as the primary data point.

Current Depth	Reports the current number of messages on queue.
Queue Depth High Event	Reports the on/off value of these events. Queue depth high events indicate that the queue depth has increased to a predefined threshold.
Queue Depth Low Event	Reports the on/off value of these events. Queue depth low events indicate that the queue depth has decreased to a predefined threshold.
Queue Depth Max Event	Reports the on/off value of these events. Queue depth max events indicate that the queue has reached its maximum depth, that is, the queue is full.
Queue Service Interval Event	Reports the on/off value of these events. Queue service interval events are related to whether messages are processed within a user-specified time interval.

Primary Data Point

The primary data point is the one of the following statistics as specified by the Statistics parameter:

- Current Depth
- Queue Depth High Event
- Queue Depth Low Event
- Queue Depth Max Event
- Queue Service Interval Event

Intelligent Data Point

The intelligent data point lists the queue manager, queue, and queue statistics. This counter reports the current state of the statistics, it does not report the statistics values as they progress through time. For dynamic information, use the Queue Manger Events counter. As appropriate, any of the following information may be included:

Data	Description
Inhibit Get = <integer>	Indicates whether get operations are allowed on the queue.
Inhibit Put = <integer>	Indicates whether put operations are allowed on the queue.

Current Queue Depth = <integer>	Current number of messages on the queue.
Maximum Queue Depth = <integer>	Maximum number of messages allowed on the queue.
Queue Depth High Event = <integer>	Variable that stores the on/off value of these events. Queue depth high events indicate that the queue depth has increased to a predefined threshold.
Queue Depth High Limit = <integer>	Value that triggers an event if it is reached.
Queue Depth Low Event = <integer>	Variable that stores the on/off value of these events. Queue depth low events indicate that the queue depth has decreased to a predefined threshold.
Queue Depth Low Limit = <integer>	Value that triggers an event if it is reached.
Queue Depth Max Event = <integer>	Variable that stores the on/off value of these events. Queue depth max events indicate that the queue has reached its maximum depth, that is, the queue is full.
Queue Service Interval Event = <integer>	Variable that stores the on/off value of these events. Queue service interval events are related to whether messages are processed within a user-specified time interval.
Queue Service Interval = <integer>	Queue service interval time.
Trigger Data = <wstring>	Free-format data that is written into a trigger message.
Trigger Depth = <integer>	Number of messages that have to be on the queue before a trigger message is written.
Trigger Control = <integer>	Controls whether or not trigger messages are written to an initiation queue.

Interval

Recommended minimum is 5 minutes.

WMI Counters

WMI Remote Extended Counters

The following extended WMI (Windows Management Instrument) remote counters are provided in QALoad. To display and use the extended counters in task configuration, you must configure user access with the MMC (Microsoft Management Console) and configure the WMI agent using the ServerVantage Agent Console (Reconfigure Agent). These procedures are described in the topic Configuring WMI in the ServerVantage Agent Configuration online help. Once configuration is complete, and you select WMI collector as your Server Type during task configuration on the Select Counters page, ServerVantage discovers the Windows registry counters and the extended counters for each WMI-configured server.

These counters extend the monitoring of your WMI system:

WMI WQL

WMI Top Ten Counters:

- CPU Utilization % - Top Ten
- Memory Utilization % - Top Ten
- I/O Utilization % - Top Ten

WMI Top Ten Counters

- Top Ten CPU
- Top Ten Memory
- Top Ten I/O

CPU Utilization % - Top Ten

The CPU Utilization % - Top Ten counter provides data for the Load Characterization Report. It returns a numeric value for each of the top ten processes that utilize the most machine CPU or all processes for which CPU utilization is greater than 0.01% at a particular moment of time.

This counter does not generate events.

Parameter

The Process parameter is not modifiable. Its value is an * (asterisk), which monitors all processes.

DataPoints

The datapoints are viewable (see above counter description).

Memory Utilization % - Top Ten

The Memory Utilization % - Top Ten provides data for Load Characterization Report. It returns a numeric value for each of the top ten processes that utilize the most machine Memory or all processes for which Memory utilization is greater than 0.01% at a particular moment of time.

This counter does not generate events.

Parameter

The Process parameter is not modifiable. Its value is an * (asterisk), which monitors all processes.

DataPoints

The datapoints are viewable (see above counter description).

I/O Utilization % - Top Ten

The I/O Utilization % - Top Ten provides data for Load Characterization Report. It returns a numeric value for each of the top ten processes that utilize the most machine I/O or all processes for which I/O utilization is greater than 0.01% at a particular moment of time.

This counter does not generate events.

Parameter

The Process parameter is not modifiable. Its value is an * (asterisk), which monitors all processes.

DataPoints

The datapoints are viewable (see above counter description).

WMI WQL

The WMI WQL (Windows Query Language) counter monitors the object (s) specified by the WQL statement. Users may select predefined WQL templates.

Parameters

WQL Statement

Enter a valid WQL (WMI Query Language) statement.

Data Point

Primary Data Point

The primary data point returns 0 if the WMI system executed query is successful. If the query fails, the graph displays DATA_NOT_FOUND as the data point. If you click on the data point, the actual error is provided in the error description.

Intelligent Data Point

The intelligent data point (IDP) is the response from the query.

Interval

Recommended minimum interval 5 minutes.

Oracle Application Server Counters

Oracle AS Counters

ServerVantage provides the following dynamically discovered Oracle Application Server (AS) remote extended counter categories for remote monitoring of Oracle10g Application Server performance metrics. Each category provides counters and parameters that extend the monitoring of your Oracle AS system. The Oracle AS agent dynamically discovers all available counters and parameter values. The available categories and metrics vary by installation. The Oracle AS agent supports wild-carded parameters and resource blackouts.

Supported platforms for Oracle AS include:

- Solaris
- AIX
- HP
- Linux
- Microsoft Windows 2000 with Service Pack 3 or above
- Microsoft Windows Server 2003 (32-bit)
- Microsoft Windows XP (not all components are supported)

Oracle AS Counter Categories

Oracle AS EJB Method Metrics	Oracle AS JMS Session Metrics
Oracle AS Entity Bean Metrics	Oracle AS JMS Store Metrics
Oracle AS HTTP OC4J Metrics	Oracle AS JMS Temp Destination Metrics
Oracle AS HTTP Server Metrics	Oracle AS JServ JSP Metrics
Oracle AS HTTP Server Module Metrics	Oracle AS JServ Metrics

Oracle AS HTTP Server Response Metrics	Oracle AS JServ Servlet Metrics
Oracle AS HTTP Server Virtual Host Metrics	Oracle AS JServ Zone Metrics
Oracle AS JDBC Connection Metrics	Oracle AS JSP Metrics
Oracle AS JDBC Connection Source Metrics	Oracle AS JVM Metrics
Oracle AS JDBC Metrics	Oracle AS Notification Server Metrics
Oracle AS JDBC Statement Metrics	Oracle AS OC4J Transaction Manager Metrics
Oracle AS JMS Browser Metrics	Oracle AS PLSQL Metrics
Oracle AS JMS Connection Metrics	Oracle AS Portal Engine Metrics
Oracle AS JMS Consumer Metrics	Oracle AS Process Manager Metrics
Oracle AS JMS Durable Subscription Metrics	Oracle AS Servlet Metrics
Oracle AS JMS Metrics	Oracle AS Task Manager Metrics
Oracle AS JMS Persistence Metrics	Oracle AS Web Module Metrics
Oracle AS JMS Producer Metrics	

10g Release 2 Counter Categories

Oracle AS Portal Cache Metrics	Oracle AS Portal Page Metrics
Oracle AS Portal Cache Summary Metrics	Oracle AS Portal DB Repository Metrics
Oracle AS Portal DB Provider Metrics	Oracle AS Portal Web Provider Metrics

Oracle Application Server Entity Bean Metrics

The Oracle Application Server (AS) Entity Bean Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
exclusive-write-access	Possible values: true or false.	Value	

Oracle Application Server HTTP OC4J Metrics

The Oracle Application Server (AS) HTTP OC4J Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
ErrReq	Specifies the total number of requests, both session and non-session, that mod_oc4j failed to route to an OC4J.	Count	Operations
ErrReqNonSess	Specifies the total number of non-session requests that mod_oc4j failed to route to an OC4J process.	Count	Operations
ErrReqSess	Specifies the total number of session requests that mod_oc4j failed to route to an OC4J process.	Count	Operations

Failover	Specifies the total number of failovers for both non-session and session requests.	Count	Operations
JVMCnt	Specifies the total number of routed OC4J JVMs that belong to this destination.	Value	Number of JVMs
NonSessFailover	Specifies the total number of failovers for non-session requests.	Count	Operations
SessFailover	Specifies the total number of failovers.	Count	Operations
SucReq	Specifies the total number of requests, both session and non-session, that mod_oc4j successfully routed to an OC4J.	Count	Operations
SucReqNonSess	Specifies the total number of non-session requests that mod_oc4j successfully routed to an OC4J process.	Count	Operations
SucReqSess	Specifies the total number of session requests that mod_oc4j successfully routed to an OC4J process.	Count	Operations
ErrReq	Specifies the total number of requests, both session and non-session, that mod_oc4j failed to route to an OC4J.	Count	Operations
ErrReqNonSess	Specifies the total number of non-session requests that mod_oc4j failed to route to an oc4j process.	Count	Operations
ErrReqSess	Specifies the total number of session requests that mod_oc4j failed to route to an OC4J process.	Count	Operations
Failover	Specifies the total number of failovers for both non-session and session requests.	Count	Operations
NonSessFailover	Specifies the total number of failovers for non-session requests. For example, assume that this mount point was serviced by an OC4J Island with three JVM's (JVM1, JVM2 and JVM3). A new non-session request is routed to JVM1. JVM1 fails to service the request, and the request is failed over to JVM2. JVM2 fails to service the request, and so the request is failed over to JVM3. At this point the NonSessFailover.Count is incremented by 2.		Operations
SessFailover	Specifies the total number of failovers for session requests. For example, assume that this mount point was serviced by an OC4J Island with three JVM's (JVM1, JVM2 and JVM3). A session request is routed to JVM1. JVM1 fails to service the request. So, the request is failed over to JVM2. At this point	Count	Operations

	the SessFailover.Count is incremented by 1. JVM2 fails to service the request, and so the request is failed over to JVM3. At this point the SessFailover.Count is incremented by 2.		
SucReqNonSess	Specifies the total number of requests, both session and non-session, that mod_oc4j successfully routed to an OC4J instance.	Count	Operations
SucReqSess	Specifies the total number of session requests that mod_oc4j successfully routed to an OC4J process.	Count	Operations
IncorrectReqInit	Total number of times an internal error occurred. There could be a number of reasons, including mod_oc4j not finding a connection endpoint and configuration errors.	Count	Operations
Oc4jUnavailable	Total number of times that an oc4j JVM could not be found to service requests.	Count	Operations
UnableToHandleReq	Total number of times mod_oc4j declined to handle a request.	Count	Operations

Oracle Application Server HTTP Server Metrics

The Oracle Application Server (AS) HTTP Server Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
busyChildren	Number of child processes active.	Value	
childFinish	Number of child processes that finish.	Count	
childStart	Number of child processes that start.	Count	
connection.active	Number of connections currently open.	Number	Threads
connection.avg	Average time spent servicing HTTP connections.	Value	Microseconds
connection.maxTime	Maximum time spent servicing any HTTP connection.		Microseconds
connection.minTime	Minimum time spent servicing any HTTP connection.		Microseconds
connection.time	Total time spent servicing HTTP connections.		Microseconds
error		Count	
get		Count	
handle.active	Child servers currently in the handle		Threads

	processing phase.		
handle.avg	Average time spent in module handler.		Microseconds
handle.completed	Number of times the handle processing phase has completed.		Operations
handle.maxTime	Maximum time spent in module handler.		Microseconds
handle.minTime	Minimum time spent in module handler.		Microseconds
handle.time	Total time spent in module handler.		Microseconds
internalRedirect	Number of times a module redirected a request to a new, internal URI.	Count	Operations
lastConfigChange		Value	
numChildren	Number of child processes.	Value	
numMods	Number of loaded modules.	Value	Operations
post		Count	
readyChildren		Value	
request.active	Child servers currently in the request processing phase.		Threads
request.avg	Average time required to service an HTTP request.		Microseconds
request.completed	Number of HTTP request completed.		Operations
request.maxTime	Maximum time required to service an HTTP request.		Microseconds
request.minTime	Minimum time required to service an HTTP request.		Microseconds
request.time	Total time required to service HTTP requests.		Microseconds
responseSize		Value	

Oracle Application Server HTTP Server Module Metrics

The Oracle Application Server (AS) HTTP Server Module Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
decline	Number of requests declined.	Count	Operations
handle.active	Number of requests currently being	Active	Requests

	handled by this module.		
handle.avg	Average time required for this module.		Microseconds
handle.completed	Number of requests handled by this module.		Operations
handle.maxTime	Maximum time required for this module.		Microseconds
handle.minTime	Minimum time required for this module.		Microseconds
handle.time	Total time required for this module.		Microseconds

Oracle Application Server HTTP Server Responses Metrics

The Oracle Application Server (AS) HTTP Server Responses Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
Response	Number of times the response was generated for each HTTP response type.	Count	

Oracle AS HTTP Server Virtual Host Metrics

The Oracle Application Server (AS) HTTP Server Virtual Host Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
request.active	Active requests.		Threads
request.avg	Average time for request processing.		Microseconds
request.completed	Number of completed requests.		Operations
request.maxTime	Maximum time to complete a request.		Microseconds
request.minTime	Minimum time to complete a request.		Microseconds
request.time			Microseconds
responseSize		Value	Bytes
vhostType		Value	

Oracle Application Server JDBC Connection Metrics

The Oracle Application Server (AS) JDBC Connection Metrics category includes the counters listed in the following table.

The parent parameter you select for these counters determines whether you get totals or data source-specific metrics.

Counters	Description	Type	Units
CreateNewStatement.avg	Average time spent creating a new statement.		Milliseconds
CreateNewStatement.completed	Number of times a request for a statement failed to be satisfied from the cache.		Operations
CreateNewStatement.maxTime	Maximum time spent creating a new statement.		Milliseconds
CreateNewStatement.minTime	Minimum time spent creating a new statement.		Milliseconds
CreateNewStatement.time	Time spent creating a new statement (this does not include the time required to parse the statement).		Milliseconds
CreateStatement.avg	Average time spent getting a statement from the statement cache.		Milliseconds
CreateStatement.completed	Number of times a request for a statement was satisfied from the cache.		Operations
CreateStatement.maxTime	Maximum time spent getting a statement from the statement cache.		Milliseconds
CreateStatement.minTime	Minimum time spent getting a statement from the statement cache.		Milliseconds
CreateStatement.time	Time spent getting a statement from the statement cache.		Milliseconds
StatementCacheHit	Statement found in cache.	Count	Operations
StatementCacheMiss	Statement not found in cache.	Count	Operations

Oracle Application Server JDBC Connection Source Metrics

The Oracle Application Server (AS) JDBC Connection Source Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
CacheFreeSize	Number of free slots in the connection cache.		Operations
CacheFreeSize.maxValue	Maximum number of free slots in the connection cache.		Connections

CacheFreeSize.minValue	Minimum number of free slots in the connection cache.		Connections
CacheFreeSize	Number of free slots in the connection cache.		Connections
CacheGetConnection.active			Threads
CacheGetConnection.avg	Average time spent getting a connection from the cache.		Milliseconds
CacheGetConnection.completed	Number of times this PhaseEvent has started and ended.		Operations
CacheGetConnection.maxTime	Maximum time spent getting a connection from the cache.		Milliseconds
CacheGetConnection.minTime	Minimum time spent getting a connection from the cache.		Milliseconds
CacheGetConnection.time	Time spent getting a connection from the cache or not.		Milliseconds
CacheHit	Number of times a request for a connection has been satisfied from the cache.	Count	
CacheMiss	Number of times a request for a connection failed to be satisfied from the cache.	Count	
CacheSize	Total size of the connection cache.	Value	

Oracle Application Server JDBC Metrics

The Oracle Application Server (AS) JDBC Metrics category includes the counters listed in the following table.

JDBC data source metrics are only available for non-emulated data sources. You are only able to access JDBC data source metrics if the data source you created is for a non-emulated data source, including OrionCMTDataSource and OracleXADataSource.

Counters	Description	Type	Units
ConnectionCloseCount	Total number of connections that have been closed.	Count	Operations
ConnectionCreate.active	Current number of threads creating connections.		Operations
ConnectionCreate.avg	Average time spent creating connections.		Milliseconds
ConnectionCreate.completed	Number of times this PhaseEvent has started and ended.		Operations

ConnectionCreate.maxTime	Maximum time spent creating connections.		Milliseconds
ConnectionCreate.minTime	Minimum time spent creating connections.		Milliseconds
ConnectionCreate.time	Time spent creating connections.		Milliseconds
ConnectionOpenCount	Total number of connections that have been opened.	Count	Operations

Oracle Application Server JDBC Statement Metrics

The Oracle Application Server (AS) JDBC Statement Metrics category includes the counters listed in the following table.

The JDBC Statement Metrics are only available for JDBC connections that have enabled statement caching and set the property `oracle.jdbc.DMSStatementCachingMetrics` to the value true. When JDBC statement caching is disabled, you can make the JDBC statement metrics available by setting the property `oracle.jdbc.DMSStatementMetrics` to true. To improve performance and to avoid collecting expensive metrics, by default these properties are both set to false.

The parent parameter you select for these counters determines whether you get totals or data source-specific metrics.

Counters	Description	Type	Units
Execute	The time this statement has spent executing the SQL including the first fetch and the time required to parse the statement.	Time	Milliseconds
Fetch	The time this statement has spent in other fetches.	Time	Milliseconds

Oracle Application Server JMS Browser Metrics

The Oracle Application Server (AS) JMS Browser Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
startTime	System.currentTimeMillis() when this browser was created.	ctor	Milliseconds
method-name	Interval timer metric (PhaseEvent Sensor) for every major method call in this browser object; calls to hasMoreElement and nextElement are made on individual enumeration objects, but counted as PhaseEvents in the browser object to simplify data collection. Multiple enumerations can be active on the same browser.	Normal	

Oracle Application Server JMS Connection Metrics

The Oracle AS JMS Connection Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
isLocal	Value is True when the JMS connection is local to the OC4J JMS server in the same JVM.	Value	Boolean
isXA	Value is True when the connection is in XA mode.	Value	Boolean
port	Remote JMS server port for this connection; set only for non-local connections.	Value	Integer
startTime	System.currentTimeMillis() when this connection was created.	Value	Milliseconds
method-name	Interval timer metric (PhaseEvent Sensor) for every major method call in this connection object.	Normal	

Oracle Application Server JMS Consumer Metrics

The Oracle Application Server (AS) JMS Consumer Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
noLocal	The noLocal setting of a subscription; set only for topic consumers.	Value	Boolean
startTime	System.currentTimeMillis() when this consumer was created.	Value	Milliseconds
method-name	Interval timer metric (PhaseEvent Sensor) for every major method call in this consumer object.	Normal	

Oracle Application Server JMS Durable Subscription Metrics

The Oracle Application Server (AS) JMS Durable Subscription Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
isActive	Value is True when the durable subscription is currently active (being used by a consumer).	Value	Boolean

noLocal	The noLocal flag for this durable subscription.	Value	Boolean

Oracle Application Server JMS Metrics

The Oracle Application Server (AS) JMS Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
connections	Number of JMS connections (local and remote) created by the JMS server.	Normal	Operations
debug	oc4j.jms.debug OC4J JMS control knob value.	ctor	Boolean
forceRecovery	oc4j.jms.forceRecovery OC4J JMS control knob value.	ctor	Boolean
listenerAttempts	oc4j.jms.listenerAttempts OC4J JMS control knob value.	ctor	Integer
maxOpenFiles	oc4j.jms.maxOpenFiles OC4J JMS control knob value.	ctor	Integer
maxOpenFiles	oc4j.jms.maxOpenFiles OC4J JMS control knob value.	ctor	Integer
messagePoll	oc4j.jms.messagePoll OC4J JMS control knob value.	ctor	Boolean
noDms	oc4j.jms.noDms OC4J JMS control knob value.	ctor	Boolean
saveAllExpired	oc4j.jms.saveAllExpired OC4J JMS control knob value.	ctor	Milliseconds
serverPoll	oc4j.jms.serverPoll OC4J JMS control knob value.	ctor	Integer
socketBufsize	oc4j.jms.socketBufsize OC4J JMS control knob value.	ctor	Boolean
usePersistence	oc4j.jms.usePersistence OC4J JMS control knob value.	ctor	Boolean
useUUID	oc4j.jms.useUUID OC4J JMS control knob value.	ctor	Integer
port	TCP/IP port on which the JMS server listens for incoming connections.	ctor	Integer
requestHandlers.count	Number of request handlers created by the JMS server.	Normal	Integer
startTime.value	System.currentTimeMillis() when the OC4J JMS server was started.	ctor	Milliseconds

taskManagerInterval	Scheduling interval of the OC4J task manager (and the scheduling interval for the OC4J JMS expiration task).	ctor	Milliseconds
method-name	Interval timer metric (PhaseEvent Sensor) for every major method call in the OC4J JMS server.	Normal	

Oracle Application Server JMS Persistence Metrics

The Oracle Application Server (AS) JMS Persistence Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
holePageCount	Number of 512b pages currently free in this file.	Normal	Integer
isOpen	Value is True when the persistence file descriptor is currently open (for LRU caching).	Normal	Boolean
lastUsed	System.currentTimeMillis() when this persistence file was last used (for LRU caching).	Normal	Milliseconds
usedPageCount	Number of 512b pages currently in use in this file.	Normal	Integer
method-name	Interval timer metric (PhaseEvent Sensor) for every major method call in the persistence file object.	Normal	

Oracle Application Server JMS Producer Metrics

The Oracle Application Server (AS) JMS Producer Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
disableMessageID	Value is true when message IDs are disabled for the producer.	Normal	Boolean
disableMessageTimestamp	Value is true when message timestamps are disabled for the producer.	Normal	Boolean
priority	Current priority of this producer.	Normal	Integer
startTime	System.currentTimeMillis() when this producer was created.	ctor	Milliseconds
timeToLive	Current timeToLive of this producer.	Normal	Milliseconds

method-name	Phase timer (PhaseEvent Sensor) metric for every major method call in this producer object.	Normal	

Oracle Application Server JMS Session Metrics

The Oracle Application Server (AS) JMS Session Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
isXA	Value is True when the session is in XA mode.	ctor	Boolean
startTime	System.currentTimeMillis() when this session was created.	ctor	Milliseconds
transacted	Value is True when the session is transacted.	ctor	Boolean
txid	Integer count of the current local transaction associated with this session; the counter is incremented each time a local transaction is committed or rolledback. Not set for non-transacted session.	Normal	Integer
method-name	Interval timer metric (PhaseEvent Sensor) for every major method call in this session object.	Normal	

Oracle Application Server JMS Store Metrics

The Oracle Application Server (AS) JMS Store Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
messageCount	Total number of messages contained in this store.	Value	Integer
messageDequeued	Total number of message dequeues (transacted or otherwise).	Count	Operations
messageDiscarded	Total number of messages discarded after the rollback of an enqueue.	Count	Operations
messageEnqueued	Total number of message enqueues (transacted or otherwise).	Count	Operations
messageExpired	Total number of message expirations.	Count	Operations
messagePagedIn	Total number of message bodies	Count	Operations

	paged in.		
messagePagedOut	Total number of message bodies paged out.	Count	Operations
messageRecovered	Total number of messages recovered (either from a persistence file, or after the rollback of a dequeue).	Count	Operations
pendingMessageCount	Total number of messages that are part of an enqueue or dequeue of an active transaction.	Value	Integer
storeSize	Total size, in bytes, of the message store.	Value	Bytes
method-name	Interval timer metric (PhaseEvent Sensor) for every major method call in the message store object.	Normal	

Oracle Application Server JMS Temp Destination Metrics

The Oracle Application Server (AS) JMS Temp Destination Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
method-name	Interval timer metric (PhaseEvent Sensor) for every major method call in the destination object.	Normal	

Oracle Application Server JServ JSP Metrics

The Oracle Application Server (AS) JServ JSP Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
processRequest.active	Threads currently in the processRequest processing phase.		Integer
processRequest.avg	Average time to completely process servlet (including JServ engine overhead).		Milliseconds
processRequest.maxTime	Maximum time to completely process servlet (including JServ engine overhead).		Milliseconds
processRequest.minTime	Minimum time to completely process servlet (including JServ engine overhead).		Milliseconds
processRequest.completed	Number of times the processRequest		Operations

	processing phase has completed.		
processRequest.time	Total time to completely process servlet (including JServ engine overhead).		Milliseconds
serviceRequest.active	Average time for service method implementing this application (excluding JServ engine overhead).		Integer
serviceRequest.avg	Average time for service method implementing this application (excluding JServ engine overhead).		Milliseconds
serviceRequest.maxTime	Maximum time for service method implementing this application (excluding JServ engine overhead).		Milliseconds
serviceRequest.minTime	Minimum time for service method implementing this application (excluding JServ engine overhead).		Milliseconds
serviceRequest.completed	Number of times the serviceRequest processing phase has completed.		Operations
serviceRequest.time	Total time for service method implementing this application (excluding JServ engine overhead).		Milliseconds
localServlet.avg	Average time to load servlet (from cache or file).		Milliseconds
localServlet.maxTime	Maximum time to load servlet (from cache or file).		Milliseconds
localServlet.minTime	Minimum time to load servlet (from cache or file).		Milliseconds
localServlet.completed	Number of times the loadServlet processing phase has completed.		Operations
localServlet.time	Total time to load servlet (from cache or file).		Milliseconds
localServletClasses.active	Threads currently in the loadServletClasses processing phase.		Count
localServletClasses.avg	Average time to load servlet classes from file.		Milliseconds
localServletClasses.maxTime	Maximum time to load servlet classes from file.		Milliseconds
localServletClasses.minTime	Minimum time to load servlet classes from file.		Milliseconds
localServletClasses.completed	Number of times the		Operations

	loadServletClasses processing phase has completed. For most classes, this value is usually one (1).		
localServletClasses.time	Total time to load servlet classes from file.		Milliseconds
loadServlet.avg	Average time to load servlet (from cache or file).		Milliseconds
createSession.active	Threads currently in the createSession processing phase.		Count
createSession.avg	Average time to create a session.		Milliseconds
createSession.maxTime	Maximum time to create a session.		Milliseconds
createSession.minTime	Minimum time to create a session.		Milliseconds
createSession.completed	Number of times the createSession processing phase has completed. Number of sessions that have been created for this application.		Operations
createSession.time	Total time to create a session.		Milliseconds
maxSTMInstances.value	Total number of instances available for this SingleThreadModel servlet.		Instances
activeSTMInstances.maxValue	Maximum number of instances concurrently servicing requests for this SingleThreadModel.		Instances
activeSTMInstances.value	Total number of instances available for this SingleThreadModel servlet.		Instances

Oracle Application Server JServ Metrics

The Oracle Application Server (AS) JServ Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
readRequest.active	Threads currently in the readRequest processing phase.		Count
readRequest.avg	Average time to read and parse requests.		Milliseconds
readRequest.maxTime	Maximum time to read and parse requests.		Milliseconds
readRequest.minTime	Minimum time to read and parse requests.		Milliseconds
readRequest.completed	Number of times the readRequest		Operations

	processing phase has completed.		
readRequest.time	Total time to read and parse the request.		Milliseconds
maxConnections	Number of requests that can be handled concurrently in the JServ process.	Value	Threads
activeConnections.maxValue	Maximum number of requests being processed simultaneously.		Threads
activeConnections	Number of requests being processed simultaneously.	Value	Threads
idlePeriod.maxTime	Maximum time process was not handling any requests.		Milliseconds
idlePeriod.minTime	Number of times no requests were being serviced.		Milliseconds
idlePeriod.completed	Number of times no requests were being serviced.		Operations
idlePeriod.time	Total time process was not handling any requests.		Milliseconds
maxBacklog	Maximum number of backlog requests that may be queued in the OS waiting for this JServ.	Value	Integer

Oracle Application Server JServ Servlet Metrics

The Oracle Application Server (AS) JServ Servlet Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
processRequest.active	Threads currently in the processRequest processing phase.		Integer
processRequest.avg	Average time to completely process servlet (including JServ engine overhead).		Milliseconds
processRequest.maxTime	Maximum time to completely process servlet (including JServ engine overhead).		Milliseconds
processRequest.minTime	Minimum time to completely process servlet (including JServ engine overhead).		Milliseconds
processRequest.completed	Number of times the processRequest processing phase has completed.		Operations

processRequest.time	Total time to completely process servlet (including JServ engine overhead).		Milliseconds
serviceRequest.active	Threads currently in the serviceRequest processing phase.		Integer
serviceRequest.avg	Average time for service method implementing this application (excluding JServ engine overhead).		Milliseconds
serviceRequest.maxTime	Maximum time for service method implementing this application (excluding JServ engine overhead).		Milliseconds
serviceRequest.minTime	Minimum time for service method implementing this application (excluding JServ engine overhead).		Milliseconds
serviceRequest.completed	Number of times the serviceRequest processing phase has completed.		Operations
serviceRequest.time	Total time for service method implementing this application (excluding JServ engine overhead).		Milliseconds
loadServlet.avg	Average time to load servlet (from cache or file).		Milliseconds
loadServlet.maxTime	Maximum time to load servlet (from cache or file).		Milliseconds
loadServlet.minTime	Minimum time to load servlet (from cache or file).		Milliseconds
loadServlet.completed	Number of times the loadServlet processing phase has completed.		Operations
loadServlet.time	Total time to load servlet (from cache or file).		Milliseconds
loadServletClasses.active	Threads currently in the loadServletClasses processing phase.		Integer
loadServletClasses.avg	Average time to load servlet classes from file.		Milliseconds
loadServletClasses.maxTime	Maximum time to load servlet classes from file.		Milliseconds
loadServletClasses.minTime	Minimum time to load servlet classes from file.		Milliseconds
loadServletClasses.completed	Number of times the loadServletClasses processing phase has completed. For most classes, this		Operations

	value is usually one (1).		
loadServletClasses.time	Total time to load servlet classes from file.		Milliseconds
loadServlet.avg	Average time to load servlet (from cache or file).		Milliseconds
createSession.active	Threads currently in the createSession processing phase.		Count
createSession.avg	Average time to create a session.		Milliseconds
createSession.maxTime	Maximum time to create a session.		Milliseconds
createSession.minTime	Minimum time to create a session.		Milliseconds
createSession.completed	Number of times the createSession processing phase has completed. Number of sessions that have been created for this application.		Operations
createSession.time	Total time to create a session.		Milliseconds
maxSTMInstances.value	Total number of instances available for this SingleThreadModel servlet.		Integer
activeSTMInstances.maxValue	Maximum number of instances concurrently servicing requests for this SingleThreadModel.		Integer
activeSTMInstances.value	Total number of instances available for this SingleThreadModel servlet.		Instances

Oracle Application Server JServ Zone Metrics

The Oracle Application Server (AS) JServ Zone Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
checkReload.active	Threads currently in the checkReload processing phase.		Integer
checkReload.avg	Average time to check if the zone must be reloaded.		Milliseconds
checkReload.maxTime	Maximum time to check if the zone must be reloaded.		Milliseconds
checkReload.minTime	Minimum time to check if the zone must be reloaded.		Milliseconds
checkReload.completed	Number of times the checkReload processing phase has completed.		Operations

checkReload.time	Total time to check if the zone must be reloaded.		Milliseconds
activeSessions	Number of times session data has been read with HttpSession.getValue in this zone.	Value	Sessions
readSession	Number of times session data has been read with HttpSession.getValue in this zone.	Count	Operations
writeSession	Number of times session data has been written with HttpSession.putValue in this zone.	Count	Operations
loadFailed	Number of times Oracle failed to load the requested application (does not work for OJSPs).	Count	Operations

Oracle Application Server JSP Metrics

The Oracle Application Server (AS) JSP Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
processRequest.time	Time spent processing requests for JSPs.		Milliseconds
processRequest.completed	Number of requests for JSPs processed by this application.		Operations
processRequest.minTime	Minimum time spent processing requests for JSPs.		Milliseconds
processRequest.maxTime	Maximum time spent processing requests for JSPs.		Milliseconds
processRequest.avg	Average time spent processing requests for JSPs.		Milliseconds
processRequest.active	Current number of active requests for JSPs.		Operations
activeInstances.value	Number of active instances. Only used when threadsafe=false.	Count	Instances
availableInstances.value	Number of available (that is, created) instances.	Count	Instances
service.active	Current number of active requests for the JSP.	Count	
service.avg	Average time spent servicing the JSP.		Milliseconds
service.completed	Number of requests for JSPs processed by this JSP.		Operations

service.maxTime	Maximum time spent servicing the JSP.		Milliseconds
service.minTime	Minimum time spent servicing the JSP.		Milliseconds
service.time	Time to serve a JSP (that is, actual execution time of the JSP).		Milliseconds

Oracle Application Server JVM Metrics

The Oracle Application Server (AS) JVM Method Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
activeThreadGroups	Number of active thread groups in the JVM.		Integer
activeThreadGroups.minValue	Minimum number of active thread groups in the JVM.		Integer
activeThreadGroupsmaxValue	Maximum number of active thread groups in the JVM.		Integer
activeThreads	Number of active threads in the JVM.		Threads
activeThreads.minValue	Minimum number of active threads in the JVM.		Threads
activeThreads.maxValue	Maximum number of active threads in the JVM.		Threads
freeMemory	Amount of heap space free in the JVM.		Kilobytes
freeMemory.minValue	Minimum amount of heap space free in the JVM.		Kilobytes
freeMemory.maxValue	Maximum amount of heap space free in the JVM.		Kilobytes
totalMemory	Total amount of heap space in the JVM.		Kilobytes
totalMemory.minValue	Minimum amount of total heap space in the JVM.		Kilobytes
totalMemory.maxValue	Maximum amount of total heap space in the JVM.		Kilobytes

Oracle Application Server Notification Server Metrics

The Oracle Application Server (AS) Notification Server Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
notifProcessed	Number of notifications processed by ONS.	Value	Operations
notifProcessQueue	Number of notifications in the process queue.	Value	Operations
notifReceived	Number of notifications received by ONS.	Value	Operations
notifReceiveQueue	Number of notifications in the receive queue.	Value	Operations

Oracle Application Server OC4J Transaction Manager Metrics

The Oracle Application Server (AS) OC4J Transaction Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
aborted_transactions	Number of aborted transactions.	Value	Integer
committed_transactions	Number of committed transactions.	Value	Integer
open_transactions	Number of open transactions.	Value	Integer

Oracle Application Server PLSQL Metrics

The Oracle Application Server (AS) PLSQL Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
newMisses	Number of new session cache misses.	Count	Operations
staleMisses	Number of stale session cache misses.	Count	Operations
hits	Number of session cache hits.	Count	Operations
requests	Number of requests to the session cache.	Count	Operations
newMisses	Number of new content cache misses.	Count	Operations
staleMisses	Number of stale content cache misses.	Count	Operations
hits	Number of content cache hits.	Count	Operations
requests	Number of requests to the content cache.	Count	Operations

error	Number of errors that have occurred within the group	Count	Operations
connFetch.maxTime	Maximum time to fetch a connection from the pool.		Microseconds
connFetch.minTime	Minimum time to fetch a connection from the pool.		Microseconds
connFetch.avg	Average time to fetch a connection from the pool.		Microseconds
connFetch.active	Child servers currently in the pool fetch phase.		Threads
connFetch.time	Total time spent fetching connections from the pool.		Microseconds
connFetch.completed	Number of times a connection has been requested from the pool.		Operations
newMisses	Number of new connection pool misses.	Count	Operations
staleMisses	Number of stale connection pool misses.	Count	Operations
hits	Number of connection pool hits.	Count	Operations

Oracle Application Server Portal Cache Metrics

The Oracle Application Server (AS) Portal Cache Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
cacheSize	Overall size of the cache.	Value	Megabytes
dataCleanedUp.max	Maximum amount of cache data clean up.		Megabytes
dataCleanedUp.min	Minimum amount of cache data clean up.		Megabytes
dataCleanedUp.avg	Average amount of cache data clean up.		Megabytes
dataCleanedUp	Amount of cache data cleaned up in the last cleanup operation.		Megabytes
cleanup	Number of times the cache has been cleaned up.	Count	Operations
cleanupTime.min	Minimum time to clean up the cache.		Milliseconds
cleanupTime.max	Maximum time to clean up the cache.		Milliseconds

cleanupTime.avg	Average time to clean up the cache.		Milliseconds
cleanupTime	Time to clean up the cache in the last cleanup operation.		Milliseconds
cacheTime.max	Maximum time to serve content from the cache.		Milliseconds
cacheTime.min	Minimum time to serve content from the cache.		Milliseconds
cacheTime.avg	Average time to serve content from the cache.		Milliseconds
openTime	Number of times cached content has been opened.		Operations
openTime.avg	Average time to open cached content.		Milliseconds
openTime.max	Maximum time to open cached content.		Milliseconds
readTime	Number of times cached content has been read.		Operations
readTime.avg	Average time to read cached content.		Milliseconds
readTime.max	Maximum time to read cached content.		Milliseconds
writeTime	Number of times cached content has been written.		Operations
writeTime.avg	Average time to write cached content.		Milliseconds
writeTime.max	Maximum time to write cached content.		Milliseconds

Oracle Application Sever Portal Cache Summary Metrics

The Oracle Application Sever (AS) Portal Cache Summary Metrics category includes the counters listed in the following table for the user and system level content cache.

Counters	Description	Type	Units
newMisses	Number of new session cache misses.		Operations
newMisses.avg	Average time to process a new cache miss.		Milliseconds
newMisses.max	Maximum time to process a new cache miss.		Milliseconds
staleMisses	Number of stale session cache misses.		Operations
staleMisses.avg	Average time to process a stale cache miss.		Milliseconds

staleMisses.max	Maximum time to process a stale cache miss.		Milliseconds
expireHits	Number of session expire cache hits.		Operations
expireHits.avg	Average time to process an expire cache hit.		Milliseconds
expireHits.max	Maximum time to process an expire cache hit.		Milliseconds
requests	Number of requests to the session cache.		Operations

Oracle Application Server Portal DB Provider Metrics

The Oracle Application Server (AS) Portal DB Provider Metrics category includes the counters listed in the following table for the Portal Servlet Database provider requests and Portal Servlet PL/SQL portlet requests.

Counters	Description	Type	Units
cacheHits	Number of cache hits for this request.	Value	
httpXXX	Count of specific HTTP response codes for this request.	Value	Operations
executeTime.maxTime	Maximum time to make the request.		Microseconds
executeTime.minTime	Minimum time to make the request.		Microseconds
executeTime.avg	Average time to make the request.		Microseconds
executeTime.active	Threads currently in the make request phase.		Threads
executeTime.time	Total time spent making requests.		Microseconds

Oracle Application Server Portal DB Repository Metrics

The Oracle Application Server (AS) Portal DB Repository Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
connFetch.maxTime	Maximum time to fetch a connection from the pool.		Milliseconds
connFetch.minTime	Minimum time to fetch a connection from the pool.		Milliseconds
connFetch.avg	Average time to fetch a connection from the pool.		Milliseconds
connFetch.time	Total time spent fetching connections from the pool.		Milliseconds

connFetch	Number of times a connection has been requested from the pool.		Operations
newMisses	Number of new connection pool misses.		Operations
staleMisses	Number of stale connection pool misses.		Operations
hits	Number of connection pool hits.		Operations
openConn	Number of currently open connections.		Operations
openConn.max	Maximum number of open connections.		Operations
openConn.avg	Average number of open connections.		Operations
requestTime	Number of requests.		Operations
requestTime.minValue	Minimum time to service a request.		Milliseconds
requestTime.maxValue	Maximum time to service a request.		Milliseconds
requestTime.time	Accumulative time of all requests.		Milliseconds

Oracle Application Server Portal Engine Metrics

The Oracle Application Server (AS) Portal Engine Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
pageRequests	Total number of requests for Portal pages.	Value	Count
cachePageHits	Number of requests for cacheable fully assembled pages that have resulted in a cache hit.	Value	Count
cachePageRequests	Number of requests for cacheable fully assembled pages	Value	Count
pageMetadataWaitTimeAvg.value	Average time spent in the PPE internal request queue waiting for page metadata, for all requests. To obtain the average, divide the value metric by the count metric. The value is the accumulative time for all requests and the count is the number of requests made.	Value	Milliseconds
pageMetadataWaitTimeAvg.count	Number of requests made for page metadata. This metric should be used in conjunction with pageMetadataWaitTimeAvg.value to calculate the average time spent in the PPE	Count	Operations

	internal request queue.		
pageMetadataWaitTime.value	Time the last page metadata request spent in the PPE internal request queue.	Value	Milliseconds
pageMetadataWaitTime.count	Number of requests for page metadata.	Count	Operations
pageMetadataWaitTime.minValue	Minimum time spent in the PPE internal request queue waiting for page metadata to be requested.		Milliseconds
pageMetadataWaitTime.maxValue	Maximum time spent in the PPE internal request queue waiting for page metadata to be requested.		Milliseconds
pageElapsedTimeAvg.value	Average time to generate pages, including fetching the page metadata. To obtain the average, divide the value metric by the count metric. The value is the accumulative time for all requests and the count is the number of requests made	Value	Milliseconds
pageElapsedTimeAvg.count	Number of pages that had to be generated (that is, not cached). use this metric in conjunction with pageElapsedTimeAvg.value to calculate the average time to generate pages, including fetching the page metadata.	Count	Operations
pageElapsedTime.value	Time to generate the last page requested, including fetching the page metadata.	Value	Milliseconds
pageElapsedTime.count	Number of pages that had to be generated (that is, not cached).	Count	Operations
pageElapsedTime.minValue	Minimum time to generate a page, including fetching the page metadata.		Milliseconds
pageElapsedTime.maxValue	Maximum time to generate a page, including fetching the page metadata.		Milliseconds
pageMetadataFetchTimeAvg	Average time to fetch page metadata, for all requests. To obtain the average, divide the value metric by the count metric. The value is the accumulative time for all requests and the count is the number of requests made.	Value	Milliseconds
pageMetadataFetchTimeAvg	Number of requests for page metadata. Use this metric in conjunction with pageMetadataFetchTimeAvg.value to calculate the average time to fetch page metadata.	Count	Operations
pageMetadataFetchTime.value	Time to fetch page metadata, for the last request.	Value	Milliseconds

pageMetadataFetchTime.count	Number of requests for page metadata.	Count	Operations
pageMetadataFetchTime.minValue	Minimum time to fetch page metadata.		Milliseconds
pageMetadataFetchTime.maxValue	Maximum time to fetch page metadata.		Milliseconds
queueTimeout	Number of requests for Portal data that have timed out in the PPE internal request queue.	Value	Milliseconds
queueStayAvg.value	Average time all internal PPE requests spent in the PPE internal request queue. To obtain the average, divide the value is the accumulative time for all requests and the count is the number of requests made.	Value	Milliseconds
queueStayAvg.count	Number of requests added to the internal PPE request queue. Use this metric in conjunction with queueStayAvg.value to calculate the average time requests spent in the internal PPE request queue.	Count	Operations
queueStay.value	Time the last internal PPE request spent in the PPE internal request queue.	Value	Milliseconds
queueStay.count	Number of requests added to the internal PPE request queue.	Count	Operations
queueStay.minValue	Minimum time a request spent in the internal PPE request queue.		Milliseconds
queueStay.maxValue	Average length of the PPE internal request queue. To obtain the average, divide the value metric by the count metric.		Milliseconds
queueLengthAvg.value	Average length of the PPE internal request queue. To obtain the average, divide the value metric by the count metric.	Value	Milliseconds
queueLengthAvg.count	Number of requests added to the PPE internal request queue. Use this metric in conjunction with queueLengthAvg.value to calculate the average length of the PPE internal request queue.	Count	Operations
queueLength.value	Current length of the PPE internal request queue.		Milliseconds
queueLength.count	Number of requests added to the PPE internal request queue.	Count	Operations
queueLength.minValue	Minimum number of requests in the PPE internal request queue.		Milliseconds
queueLength.maxValue	Maximum number of requests in the PPE internal request queue.		Milliseconds

cacheHits	Number of cache hits for this request.	Value	Operations
httpXXX	Count of specific HTTP response codes for this request.	Value	Operations
executeTime.maxTime	Maximum time to make the request.		Microseconds
executeTime.minTime	Minimum time to make the request.		Microseconds
executeTime.avg	Average time to make the request.		Microseconds
executeTime.active	Threads currently being processed.		Threads
executeTime.time	Total time spent making requests.		Microseconds
connFetch.completed	Number of requests made.		Operations

Oracle Application Server Portal Page Metrics

The Oracle Application Server (AS) Portal Page Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
pageRequests	Total number of requests for Portal pages.	Value	Count
cachePageHits	Number of requests for cacheable fully assembled pages that have resulted in a cache hit.	Value	Count
cachePageRequests	Number of requests for cacheable fully assembled pages.	Value	Count
pageMetadataWaitTimeAvg.value	Average time spent in the PPE internal request queue waiting for page metadata, for all requests. To obtain the average, divide the value metric by the count metric. The value is the accumulative time for all requests and the count is the number of requests made.		Milliseconds
pageMetadataWaitTimeAvg.count	Number of requests made for page metadata. Use this metric in conjunction with pageMetadataWaitTimeAvg.value to calculate the average time spent in the PPE internal request queue.		Operations
pageMetadataWaitTime.value	Time the last page metadata request spent in the PPE internal request queue.	Value	Milliseconds
pageMetadataWaitTime.count	Number of requests for page metadata.	Count	Operations
pageMetadataWaitTime.minValue	Minimum time spent in the PPE internal request queue waiting for page metadata to be requested.		Milliseconds

pageMetadataWaitTime.maxValue	Maximum time spent in the PPE internal request queue waiting for page metadata to be requested.		Milliseconds
pageElapsedTimeAvg.value	Average time to generate pages, including fetching the page metadata. To obtain the average, divide the value metric by the count metric. The value is the accumulative time for all requests and the count is the number of requests made.	Value	Milliseconds
pageElapsedTimeAvg.count	Number of pages that had to be generated (that is, not cached). Use this metric in conjunction with pageElapsedTimeAvg.value to calculate the average time to generate pages, including fetching the page metadata.	Count	Operations
pageElapsedTime.value	Time to generate the last page requested, including fetching the page metadata.		Milliseconds
pageElapsedTime.count	Number of pages that had to be generated (that is, not cached).		Operations
pageElapsedTime.minValue	Minimum time to generate a page, including fetching the page metadata.		Milliseconds
pageElapsedTime.maxValue	Maximum time to generate a page, including fetching the page metadata.		Milliseconds
pageMetadataFetchTimeAvg.value	Average time to fetch page metadata, for all requests. To obtain the average you should divide the value metric by the count metric. The value being the accumulative time for all requests and the count being the number of requests made.		Milliseconds
pageMetadataFetchTimeAvg.count	Number of requests for page metadata. This metric should be used in conjunction with pageMetadataFetchTimeAvg.value to calculate the average time to fetch page metadata.		Operations
pageMetadataFetchTime.value	Time to fetch page metadata, for the last request.		Milliseconds
pageMetadataFetchTime.count	Number of requests for page metadata.		Operations
pageMetadataFetchTime.minValue	Minimum time to fetch page metadata.		Milliseconds
pageMetadataFetchTime.maxValue	Maximum time to fetch page metadata.		Milliseconds
queueTimeout	Number of requests for Portal data that have timed out in the PPE internal request queue.	Value	Milliseconds
queueStayAvg.value	Average time all internal PPE requests spent		Milliseconds

	in the PPE internal request queue. To obtain the average, divide the value metric by the count metric. The value is the accumulative time for all requests and the count is the number of requests made.		
queueStayAvg.count	Number of requests added to the internal PPE request queue. Use this metric in conjunction with queueStayAvg.value to calculate the average time requests spent in the internal PPE request queue.		Operations
queueStay.value	Time the last internal PPE request spent in the PPE internal request queue.		Milliseconds
queueStay.count	Number of requests added to the internal PPE request queue.		Operations
queueStay.minValue	Minimum time a request spent in the internal PPE request queue.		Milliseconds
queueStay.maxValue	Maximum time a request spent in the internal PPE request queue.		Milliseconds
queueLengthAvg.value	Average length of the PPE internal request queue. To obtain the average, divide the value metric by the count metric.		Milliseconds
queueLengthAvg.count	Number of requests added to the PPE internal request queue. Use this metric in conjunction with queueLengthAvg.value to calculate the average length of the PPE internal request queue.		Operations
queueLength.value	Current length of the PPE internal request queue.		Milliseconds
queueLength.count	Number of requests added to the PPE internal request queue.		Operations
queueLength.minValue	Minimum number of requests in the PPE internal request queue.		Milliseconds
queueLength.maxValue	Maximum number of requests in the PPE internal request queue.		Milliseconds
requests.	Number of page requests.	Value	Operations
httpXXX	Count of specific HTTP response codes.	Value	Operations
httpFailure	Count of internal Parallel Page Engine errors encountered whilst requesting portlets.	Value	Operations
httpTimeout	Count of timeouts encountered whilst requesting portlets.	Value	Operations
httpUnresolvedRedirect	Count of requests for portlets which resulted	Value	

	in a redirected request not being resolved successfully.		

Oracle Application Server Portal Web Provider Metrics

The Oracle Application Server (AS) Portal Web Provider Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
cacheHits	Number of cache hits for this request.	Value	Operations
httpXXX	Count of specific HTTP response codes for this request.	Value	Operations
executeTime.maxTime	Maximum time to make the request.		Microseconds
executeTime.minTime	Minimum time to make the request.		Microseconds
executeTime.avg	Average time to make the request.		Microseconds
executeTime.active	Threads currently in the make request phase.		Threads
executeTime.time	Total time spent making requests.		Microseconds

Oracle Application Server Process Manager Metrics

The Oracle Application Server (AS) Process Manager Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
jobWorkerQueue	Number of jobs in the OPMN worker queue.	Value	Operations
lReq	Number of local HTTP requests which OPMN handles.	Count	Operations
procDeath	Number of processes which die after the process manager starts them.	Count	Operations
procDeathReplace	Number of processes which are restarted after the process manager detects they are dead.	Count	Operations
reqFail	Number of HTTP requests which fail.	Count	Operations
reqPartialSucc	Number of HTTP requests which partially succeed.	Count	Operations
reqSucc	Number of HTTP requests which succeed.	Count	Operations
rReq	Number of remote HTTP requests which OPMN handles.	Count	Operations

workerThread	Number of worker threads.	Value	Threads
cpuIdle	Number of milliseconds the CPU(s) have been idle since an unspecified time.	Value	Milliseconds
freePhysicalMem	Amount of free physical memory on the host machine.	Value	Kilobytes
numProcessors	Number of processors available on the host machine.	Value	Integer
totalPhysicalMem	Total physical memory available on the host machine.	Value	Kilobytes
numProcConf	Number, or maximum number, of processes configured for this process set.	Value	Integer
reqFail	Number of HTTP requests which fail for this process set.	Count	Operations
reqPartialSucc	Number of HTTP requests which partially succeed for this process set.	Count	Operations
reqSucc	Number of HTTP requests which succeed for this process set	Count	Operations
cpuTime	Amount of CPU time used by the process.	Value	Cpu Milliseconds
heapSize	Heap size of the process.	Value	Kilobytes
privateMemory	Private memory of the process.	Value	Kilobytes
sharedMemory	Shared memory for the process.	Value	Milliseconds

Oracle Application Server Servlet Metrics

The Oracle Application Server (AS) Servlet Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
service.active	Current number of threads servicing this servlet.		Threads
service.avg	Average time spent in servicing the servlet.		Milliseconds
service.completed	Total number of calls to service.	Count	
service.maxActive	Maximum number of threads servicing this servlet.		Threads
service.maxTime	Maximum time spent on a servlet's service() call.		Operations

service.minTime	Minimum time spent on a servlet's service() call.		Milliseconds
service.time	Total time spent on the servlet's service() call.		Milliseconds

Oracle Application Server Task Manager Metrics

The Oracle Application Server (AS) Task Manager Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
interval	How often the task should run. The task manager executes all the tasks in a round-robin fashion. If the interval is zero, then the task manager executes the task when it is selected in the round robin.	Value	Milliseconds
run().active	Number of active threads.		Threads
run().avg	Average time for the task manager to run the task.		Milliseconds
run().completed	Number of times the task manager has run the task.		Operations
run().maxActive	Maximum number of active tasks.		Threads
run().maxTime	Maximum time for the task to run.		Milliseconds
run().minTime	Minimum time for the task to run.		Milliseconds
run().time	Total time spent running the task manager.		Milliseconds

Oracle Application Server Web Module Metrics

The Oracle Application Server (AS) Web Module Metrics category includes the counters listed in the following table.

Counters	Description	Type	Units
resolveServlet.time	Total time spent to create/locate servlet instances (within the servlet context). This includes the time for any required authentication.		Milliseconds
resolveServlet.completed	Total number of lookups for a servlet by OC4J.		Operations
resolveServlet.minTime	Minimum time spent to create/locate the servlet instance (within the servlet context).		Milliseconds

resolveServlet.maxTime	Maximum time spent to create/locate the servlet instance (within the servlet context).		Milliseconds
resolveServlet.avg	Average time spent to create/locate the servlet instance (within the servlet context).		Milliseconds
sessionActivation.active	Number of active sessions.		Operations
sessionActivation.time	Total time in which sessions have been active.		Milliseconds
sessionActivation.completed	Number of session activations.		Operations
sessionActivation.minTime	Minimum time a session was active.		Milliseconds
sessionActivation.maxTime	Maximum time a session was active.		Milliseconds
sessionActivation.avg	Average session lifetime.		Milliseconds
service.time	Total time spent servicing requests.		Milliseconds
service.completed	Total number of requests serviced.		Operations
service.minTime	Minimum time spent servicing requests.		Milliseconds
service.maxTime	Maximum time spent servicing requests.		Milliseconds
service.avg	Average time spent in servicing the servlet.		Milliseconds
service.active	Current number of requests active.		Operations
parseRequest.active	Current number of threads trying to read/parse AJP or HTTP requests.		
parseRequest.avg	Average time spent to read/parse requests.		Milliseconds
parseRequest.completed	Number of web requests that have been parsed.		Operations
parseRequest.maxActive	Maximum number of threads trying to read/parse AJP or HTTP requests.		Threads
parseRequest.maxTime	Maximum time spent to read/parse requests.		Milliseconds
parseRequest.minTime	Minimum time spent to read/parse requests.		Milliseconds
parseRequest.time	Total time spent to read/parse requests from the socket.		Milliseconds

processRequest.active	Current number of threads servicing web requests.		
processRequest.avg	Average time spent servicing web requests.		Milliseconds
processRequest.completed	Number of web requests processed by this application.		Operations
processRequest.maxActive	Maximum number of threads servicing web requests.		Threads
processRequest.maxTime	Maximum time spent servicing a web request.		Milliseconds
processRequest.minTime	Minimum time spent servicing a web request.		Milliseconds
processRequest.time	Total time spent servicing this application's web requests.		Milliseconds
resolveContext.active	Current number of threads trying to create/find the servlet context.		
resolveContext.avg	Average time spent to create/find the servlet context.		Milliseconds
resolveContext.completed	Count of completed context resolves.		Operations
resolveContext.maxActive	Maximum number of threads trying to create/find the servlet context.		Threads
resolveContext.maxTime	Maximum time spent to create/find the servlet context.		Milliseconds
resolveContext.minTime	Minimum time spent to create/find the servlet context.		Milliseconds
resolveContext.time	Total time spent to create/find the servlet context. Each web module (WAR) maps to a servlet context.		Milliseconds

JVM Counters

JVM Counters

ServerVantage provides the following statically discovered categories for monitoring a Java Virtual Machine (JVM). Each category provides counters that allow the monitoring of your JVM. ServerVantage utilizes the Java Monitoring and Management API which were introduced in J2SE 5.0 for counter data.

JVM Class Loading	JVM Memory
JVM Compilation	JVM Operating System
JVM Garbage Collection	JVM Threads

Guideline:

You must start your JVM as a "JMX-enabled JVM"by inserting the following properties:

```
C:\...\java -Dcom.sun.management.jmxremote.port=1095 -
Dcom.sum.management.jmxremote.ssl=false  -
Dcom.sun.management.jmxremote.authenticate=false
```

For more information, see http://java.sun.com/j2se/1.5.0/docs/guide/management/agent.html.

JVM Class Loading Counters

The JVM Class Loading category includes the counters listed in the following table. ServerVantage utilizes JMX and the Java Monitoring and Management API which were introduced in the Java Virtual Machine (JVM) 1.5, release J2SE 5.0.

Counters	Description	Type	Units
Compilation Time	Approximate time (in milliseconds) spent in compilation by the JVM during the sample interval.	Integer	Number of milliseconds
Current Loaded Class Count	Number of classes currently loaded in the JVM.	Integer	Number of classes
Loaded Class Count	Number of classes loaded since JVM started execution.	Long	Number of classes
Unloaded Class Count	Number of classes unloaded since the JVM started execution.	Long	Number of classes

Parameters

The following parameter is valid for this counter category.

JMX Port

JMX port associated with the JVM you want to monitor.

Data Point

The primary data point (PDP) is the value returned for the counter used in the task.

Interval

Recommended minimum is 5 minutes.

JVM Compilation Counters

The JVM Compilation category includes the counters listed in the following table. ServerVantage utilizes JMX and the Java Monitoring and Management API which were introduced in the Java Virtual Machine (JVM) 1.5, release J2SE 5.0.

Counters	Description	Type	Units

Compilation Time	Approximate time (in milliseconds) spent in compilation by the JVM during the sample interval.	Long	Milliseconds

Parameters

The following parameter is valid for this counter category.

JMX Port

JMX port associated with the JVM you want to monitor.

Data Point

The primary data point (PDP) is the value returned for the counter used in the task.

Interval

Recommended minimum is 5 minutes.

JVM Garbage Collection Counters

The JVM Garbage Collection category includes the counters listed in the following table. ServerVantage utilizes JMX and the Java Monitoring and Management API which were introduced in the Java Virtual Machine (JVM) 1.5, release J2SE 5.0.

Counters	Description	Type	Units
Garbage Collection Count	Number of collections that have occurred during the sample interval.	Long	Number of collections
Garbage Collection Time	Approximate time (in milliseconds) spent garbage collecting by the JVM during the sample interval.	Long	Milliseconds
Garbage Collection Count	Total number of collections that have occurred during the sample interval.	Long	Number of collections
Garbage Collection Time	Total approximate time (in milliseconds) spent garbage collecting by the JVM during the sample interval.	Long	Milliseconds

Parameters

The following parameters are valid for this counter category.

JMX Port

JMX port associated with the JVM you want to monitor.

Collector Name

This parameter is available with some of the garbage collection counters. It provides the name of the garbage collector you want to monitor. For the Hotspot JVM, the values are Copy and MarkSweepCompact.

Data Point

The primary data point (PDP) is the value returned for the counter used in the task.

Interval

Recommended minimum is 5 minutes.

JVM Memory Counters

The JVM Memory category includes the counters listed in the following table. ServerVantage utilizes JMX and the Java Monitoring and Management API which were introduced in the Java Virtual Machine (JVM) 1.5, release J2SE 5.0. -

Counters	Description	Type	Units
Collection Usage Threshold Count(M)	Number of times that the JVM has detected that the memory usage has reached or exceeded the collection usage threshold for an identified memory pool.	Integer	Number of times
Committed Memory Heap	Amount of heap memory that is committed to the JVM for use. The JVM has a heap that is the runtime data area from which memory for all class instances and arrays are allocated. It is created at the JVM start-up. Heap memory for objects is reclaimed by an automatic memory management system which is known as a garbage collector.	Long	Bytes
Committed Memory(M)	Amount of memory that is guaranteed to be available to the JVM to use for the identified memory pool.	Long	Bytes
Committed Non-heap Memory	Amount of non-heap memory that is guaranteed to be available to the JVM for use.	Long	Bytes
Maximum	Maximum amount of heap	Long	Bytes

Heap Memory	memory that can be used for memory management.		
Maximum Memory(M)	Maximum amount of memory that can be used for memory management for this memory pool.	Long	Bytes
Maximum Non-heap Memory	Maximum amount of non-heap memory that can be used for memory management.	Long	Bytes
Objects Pending Finalization Count	Approximate number of objects for which finalization is pending.	Integer	Number of objects
Peak Committed Memory(M)	Peak amount of memory (in bytes) that was guaranteed to be available for use by the JVM for the identified memory pool since the JVM was started or since the peak was reset.	Long	Bytes
Peak Maximum Memory(M)	Peak maximum amount of memory (in bytes) that was available to the JVM for the identified memory pool since the JVM was started or since the peak was reset.	Long	Bytes
Peak Used Memory(M)	Peak used memory (in bytes) for the identified memory pool since the JVM was started or since the peak was reset.	Long	Bytes
Post Collection Committed Memory(M)	Amount of memory (in bytes) that is guaranteed to be available for use by the JVM for the identified memory pool after the JVM most recently expended effort in recycling unused objects.	Long	Bytes
Post Collection Maximum Memory(M	Maximum amount of memory (in bytes) that is available to the JVM for the identified memory pool after the JVM most recently expended effort in recycling unused objects.	Long	Bytes

Post Collection Used Memory(M)	Used memory (in bytes) for the identified memory pool after the JVM most recently expended effort in recycling unused objects	Long	Bytes
Total Collection Usage Threshold Count(M)	Total number of times that the JVM has detected that the memory usage has reached or exceeded the collection usage threshold for a memory pool.	Long	Number of times
Total Committed Memory(M)	Amount of memory (in bytes) that is guaranteed to be available for use by the JVM.	Long	Bytes
Total Maximum Memory(M)	Maximum amount of memory (in bytes) available to the JVM for memory management.	Long	Bytes
Total Usage Threshold Count(M)	Number of times that the JVM has detected that the memory usage for a memory pool has reached or exceeded the usage threshold for the memory pool.	Long	Number of times
Total Used Memory(M)	Amount of memory (in bytes) currently in use by the JVM.	Long	Bytes
Usage Threshold Count(M)	Number of times that the JVM has detected that the memory usage for a memory pool has reached or exceeded the usage threshold for the memory pool.	Long	Number of times
Used Heap Memory	Amount of heap memory (in bytes) currently in use by the JVM. The JVM has a heap that is the runtime data area from which memory for all class instances and arrays are allocated. It is created at the JVM start-up. Heap memory for objects is reclaimed by an automatic memory management	Long	Bytes

	system which is known as a garbage collector.		
Used Memory(M)	Used memory (in bytes) for the identified memory pool.	Long	Bytes
Used Non-heap Memory	Amount of non-heap memory (in bytes) currently in use by the JVM.	Long	Bytes

Parameters

The following parameters are valid for this counter category.

JMX Port

JMX port associated with the JVM you want to monitor.

Memory Pool Name

This parameter is available with some of the memory pool counters. It provides the name of the memory pool you want to monitor. For the Hotspot JVM, the values are Code Cache and Survivor Space.

Data Point

The primary data point (PDP) is the value returned for the counter used in the task.

Interval

Recommended minimum is 5 minutes.

JVM Operating System Counter

The JVM Operating System category includes the counters listed in the following table. ServerVantage utilizes JMX and the Java Monitoring and Management API which were introduced in the Java Virtual Machine (JVM) 1.5, release J2SE 5.0.

Counters	Description	Type	Units
Available Processor Count	Number of processors available to the JVM.	Integer	Number of processors

Parameters

The following parameter is valid for this counter category.

JMX Port

JMX port associated with the JVM you want to monitor.

Data Point

The primary data point reports the number of processors available to the JVM .

Interval

Recommended minimum is 5 minutes.

JVM Threads Counters

The JVM Threads category includes the counters listed in the following table. ServerVantage utilizes JMX and the Java Monitoring and Management API which were introduced in the Java Virtual Machine (JVM) 1.5, release J2SE 5.0.

Counters	Description	Type	Units
Live Daemon Thread Count	Number of live daemon threads.	Integer	Number of threads
Live Thread Count	Number of live threads.	Integer	Number of threads
Monitor Deadlocked thread count	Number of threads that are in deadlock waiting to acquire object monitors. A thread is monitor-deadlocked if it is part of a cycle in the relation "is waiting for an object monitor owned by." In the simplest case, thread A is blocked waiting for a monitor owned by thread B, and thread B is blocked waiting for a monitor owned by thread A.	Integer	Number of threads
Started Thread Count	Number of threads started by the JVM during the sample interval.	Integer	Number of threads

Parameters

The following parameters are valid for this counter category.

JMX Port

JMX port associated with the JVM you want to monitor.

Thread ID

Identifies the individual thread in the process you want to monitor.

Data Point

The primary data point (PDP) is the value returned for the counter used in the task.

Interval

Recommended minimum is 5 minutes.

Managing Counters

Adding Counters to a Task Using New Discovery Data

Add counters to a monitoring task by generating the available counter data and selecting the counters and instances to add to the task.

To add counters and instances to a monitoring task:

1. From the Conductor menu bar, click Tools>Monitor Tasks to open the Manage Monitoring Tasks dialog window.
2. Open or create a task.
3. In the menu bar, click Actions>Add counter>Use new discovery counters. The Edit Existing Monitor Wizard appears.
4. Follow the instructions for using the wizard to discover and add counters to the monitoring task.

Notes:

! (WebLogic and WebSphere) When the monitor to which you are adding counters is managed by an administrative server, the Edit Existing Monitor wizard appears and the counter discovery process begins immediately.

! (SNMP) You can supplement the counter discovery list by creating and specifying a custom OID file. See Customizing SNMP Counter Discovery for more information.

Adding Counters to a Task Using Cached Discovery Data

It is possible to add counters to monitor for a machine and monitor type using cached discovery data. To add counters to a task using cached discovery, do the following:

Step 1: Select the counter to add or modify

Step 2: Choose the instances of the counter to monitor

Step 3: Review the monitor definition

Step 4: Save the task

Step 1: Select the counter to add or modify:

1. From the Conductor menu bar, click Tools>Monitor Tasks to open the Manage Monitoring Tasks dialog window.
2. In the menu bar, click Actions>Add counter>Use cached discovery counters. This Add/Edit Counters dialog box appears.
3. From the Available Items pane, select the Template tab or the Counter tab.
4. To add an item, select a template or a counter, and click Add, or double-click the item to display it in the Selected Items pane. Click Add All to add all the items on the selected tab to the Selected Items pane.
5. To remove an item, double-click the item in the Selected Items pane or select the item and click Remove. The item is returned to the Available Items pane.

Note: Select multiple counters and templates by doing one of the following:

- To select nonadjacent counter items, click a counter item, and then hold down Ctrl and click each additional counter item.
- To select adjacent counter items, click the first counter item in the sequence, and then hold down Shift and click the last counter item.

6. Click Next. The Choose Instances dialog box displays.

Note: When you select a template, and some of the counters it contains are not present on the machine you are defining, you receive a message with a list of the counters that will not be added to the task.

Step 2: Choose the instances of the counter to monitor:

When clicking Next in the previous dialog box. The Choose Instances dialog box appears.

1. Review the counters selected. When a red dot appears next to a counter, select an instance of the counter.
2. Double-click the counter group to display the counters.
3. Select a counter and click Edit. The Select instance for counter dialog box appears.
4. In the Available Instance pane, select an instance and click Add.
5. Repeat until all instances of the counter that you want to apply to the task are selected.
6. Click Save. The Choose Instances dialog box appears.
7. Repeat this process for each designated counter.
8. Click Next. The Summary dialog box displays.

Step 3: Review the monitor definition:

1. In the Review Monitor Definition dialog box, review the information for the monitoring machine you defined.
2. Select one of the following:
   - Set up another monitor for this task - returns to the Enter properties of the monitoring machine dialog box so you can add another monitor to the monitoring task. When complete, proceed with Step 3 below.
   - Continue without adding any more monitors - continues in this dialog box.

Note: (WebLogic and WebSphere) When you set up another monitor in a managed server environment, you only can add another monitor using a different administrative server.

   - To add a WebLogic monitor, you must use the same WebLogic jar files and WebLogic version as the current monitor.
   - To add a WebSphere monitor, you must use the same WebSphere Home, WebSphere client version, and WebSphere server version as the current monitor.

3. (Optional) In the Monitors pane, select the monitor type and click Save as Template to create a template for this monitoring task.
4. (Optional) In the Monitors pane, select a monitor and click Remove Monitor to delete a monitor from the task.
5. (Optional) Type a new value in the Sample Interval field. This is the frequency, in seconds, at which QALoad requests data during runtime data collection.

6. Click Next to proceed to the next step, where you review and save your selections.

Step 4: Save the task:

When clicking Next in the previous dialog box, the Summary dialog box appears.

1. On the Summary dialog box, review the monitors and counters selected for the template. Click Back to return to a dialog box and make changes to the information.
2. Click Finish to add the counters.

Removing a Monitor or a Counter from a Monitoring Task

Remove a monitor or a counter from a monitoring task, by following this procedure.

To remove a counter from a monitoring task:

1. In the Monitors pane of the Manage Monitoring Tasks window, select the monitor, counter, or counter family to delete.
2. Click Actions>Remove Monitor/Counter.
3. When the verification dialog box displays, click OK.

Note: You cannot remove the last monitor on a machine, the last counter in the family, or the last family of counters in the task.

Monitoring Templates

About Monitoring Templates

Monitoring templates are designed to facilitate the configuration process. A monitoring template is a predefined group of counters not associated with a specific machine. You can create a new template for a monitoring task, or you can use one of QALoad's pre-defined templates.

When you create a custom template, QALoad's New Monitoring Template wizard guides you through the process of defining the type of template you want to create, configuring the monitor properties, and adding the counters and instances of counters to the template.

When you use one of QALoad's predefined templates, you select a stored template with the counters you want to monitor. The templates have counters grouped by functionality, such as Network Traffic, Response Time, or System Health. Where appropriate, the templates include the specific instances to monitor for each counter.

You can add or edit counters in either custom or pre-defined templates. When you open a template to edit it for the first time, Edit Monitoring Template wizard guides you through the process of discovering and adding new counters to a template. When you've just completed the counter discovery process for a template, either by creating a new template or by opening a template for editing, you can select counters from those already available in memory by using the cached discovery.

Custom Templates

You can create templates of the monitoring tasks that you develop so that all of the counters and instances for the task are saved. You can create new tasks and incorporate the template you created. Templates are saved as .xml files in the Templates directory.

You can create a template when you define a monitoring task, or you can use the New Monitor Template wizard to create and store a template for future use. Custom templates can be modified using either new discovery data or cached discovery data.

Pre-defined Templates

About Pre-defined Templates

QALoad provides pre-defined templates for each monitor type. Each template includes the counters most commonly used for particular task within each monitor type.

QALoad provides the templates form the following monitor types:

! Oracle Application Server

! JVM

! SAP

! SNMP

! WebLogic

! WebSphere

! WebSphere MQ

! Windows Registry

! WMI

Note: You cannot modify pre-defined templates.

Viewing Pre-defined Templates

QALoad provides pre-defined templates for each monitor type. These include the counters most commonly used for particular task.

To access and review pre-defined templates:

1. In the Conductor, select Tools>Monitor Tasks to open the Manage Monitoring Tasks window.
2. In the Manage Monitoring Tasks menu bar, click Templates>Open Existing. The Select a Monitor Template File dialog box appears and lists pre-defined templates and custom templates you created and saved.
3. Double-click a monitor type, then select a template file and click Open.
4. The template name displays in the Manage Monitoring Tasks dialog box.

Oracle Application Server Templates

Oracle Application Server Template Index

QALoad provides the following pre-defined Oracle Application Server (AS) 10g database templates:

Oracle AS Availability

Oracle AS Performance

Oracle Application Server Availability

This template monitors the availability of an Oracle Application Server (AS) 10g database.

Note: This template is only available if you have Oracle AS 10g installed on your monitored server.

This template includes the following counters and categories:

Category	Counters	Description
Oracle AS HTTP OC4J Metrics	OC4JUnavailable	Total number of times that an oc4j JVM could not be found to service requests.
	UnableToHandleReq	Total number of times mod_oc4j declined to handle a request.
Oracle AS HTTP Server Metrics	readyChildren	Number of child processes that are ready to run.
Oracle AS Process Manager Metrics	reqFail	Number of HTTP requests which fail.

Oracle Application Server Performance

This template monitors the availability of an Oracle Application Server (AS) 10g database.

Note: This template is only available if you have Oracle AS 10g installed on your monitored server.

This template includes the following counters and categories:

Category	Counters	Description
Oracle AS HTTP OC4J Metrics	ErrReqSess	Specifies the total number of session requests that mod_oc4j failed to route to an OC4J process.
	SucReqSess	Specifies the total number of session requests that mod_oc4j successfully routed to an OC4J process.
Oracle AS HTTP Server Metrics	busyChildren	Number of child processes active.
	connection.avg	Average time spent servicing HTTP connections.
	request.avg	Average time required to service an HTTP request.

SAP Templates

SAP Templates

QALoad provides the following pre-defined SAP templates:

QALoad-SAP R3 Remote Availability

QALoad-SAP R3 Remote Performance

QALoad-SAP R3 Remote System Errors

QALoad-SAP R3 Remote Availability

This template monitors the availability of an SAP R/3 Instance. The SAP R/3 Availability template returns critical information about the availability of your SAP installation. One metric used to determine the availability of an SAP R/3 Instance is the status of the SAP collector.

The default event action assigned to this template issues an alarm if either the specified R/3 Instance or the collector goes down. The default instance is the first SAP Instance configured for monitoring during installation.

The SAP R/3 Availability template uses the following SAP R/3 extended counters:

Counters	Description
Active Servers	Returns the number of active SAP application servers for a given instance. It detects when a remote server is unavailable. Rule: IF 'SAP R/3 Remote Extended.Active Servers(SAP Instance: "**", Server Count: "10")' = 0 .

QALoad-SAP R3 Remote Performance

This template monitors the performance of your SAP R/3 Instance.

The default event action for this template raises an event if the number of alerts of critical status is greater than 0, or if the buffer hit ratio falls below 95%.

All the counters associated with this template require the instance number of your SAP installation. By default, this template uses the first instance configured for monitoring during ServerVantage installation. If you use the task configuration wizard to change the instance that the template monitors, you must also change the instance specified in the rule accordingly.

The SAP R/3 Performance template uses the following SAP R/3 extended counters:

Counters	Description
Buffer Statistic	Returns different buffer statistics for selected buffer name. This counter was chosen because buffering data is a key to the performance of SAP. Rule: IF 'SAP R/3 Remote Extended.Buffer Statistic(SAP Instance: "**", Buffer Name: "TTAB", Statistic Name: "Hit rate SAP buffer(%%)")' < 95.
Itemized Spool Queue	Return number of entries in the spool queue that match the specified criteria. Rule: IF 'SAP R/3 Remote Extended.Spool Queue(SAP Instance: "**", Request Status: "Processing")' > 10.
Memory Usage	Returns current memory usage. Rule: IF 'SAP R/3 Remote Extended.Memory Usage(SAP Instance: "**", Count: "10", Metrics: "MB")' > 10000.
Page/Roll Area	Returns Used Paging Area % statistic. This counter was chosen because roll memory is critical for work processes and page memory is critical for internal data processing.
Work Processes	Counter for monitoring SAP R/3 work processes. Returns number of stopped work processes. Rule: IF 'SAP R/3 Remote Extended.Work Processes(SAP Instance: "**", Process Type: "BGDDIAENQSPOUP2UPD", Process State: "Stopped")' > 2.

QALoad-SAP R3 Remote System Errors

This template monitors the errors and critical situations that occur on a SAP R/3 system. Rules and thresholds are preset to appropriate values for most sites.

The default sampling interval for this template is 5 minutes.

The SAP R/3 Performance template uses the following SAP R/3 Remote extended counters:

Counters	Description
Alerts	Counter for monitoring R/3 alerts. Returns number of alerts according to the specified criteria. This counter checks all alerts with error (red) status. Rule: IF 'SAP R/3 Remote Extended.Alerts(SAP Instance: "**", Monitor Set: "SAP CCMS Admin Workplace", Monitor: "Database", Severity: "Error - Red", Pattern: "*", Show Alert Text: "No")' > 0.
Itemized Spool Queue	Return number of entries in the spool queue that match the specified criteria.
Spool Queue	Return number of entries in the spool queue that match the specified criteria. This counter checks all spool entries with “Problem” status. Rule: IF 'SAP R/3 Remote Extended.Spool Queue(SAP Instance: "**", Request Status: "Problem")' > 0.
Work Processes	Counter for monitoring SAP R/3 work processes. Returns number of work processes according to the specified criteria. This counter checks stopped work processes. Rule: IF 'SAP R/3 Remote Extended.Work Processes(SAP Instance: "**", Process Type: "BGDDIAENQSPOUP2UPD", Process State: "Stopped")' > 0.

SNMP Templates

SNMP Templates

QALoad provides the following pre-defined SNMP templates:

QALoad-HP Performance

QALoad-Linux Performance

QALoad-SUN Performance

QALoad-HP Performance

This template includes the following counters and categories:

Category	Counters	Description
HP System	CpuIdle%	CpuSys% is the percentage of idle processor

		time.
	CpuSys%	CpuSys% is the percentage of non-idle processor time that is spent in system mode.
	CpuUser%	CpuUser% is the percentage of non-idle processor time that is spent in user mode.
	FreeMemory KBytes	FreeMemory KBytes is the amount of idle memory.
	FreeSwap KBytes	FreeSwap is the amount of free swap space on the system.
	MaxUserMem KBytes	MaxUserMem is the amount of maximum user memory on the system.
	Users	Users is the number of users logged on to the machine.
tcp	tcpInSegs/sec	tcpInSegs/sec is the rate at which segments are received, including those received in error.
	tcpOutSegs/sec	tcpOutSegs/sec is the rate at which segments are sent, including those on current connections but excluding those containing only retransmitted octets.
udp	udpInDatagrams/sec	udpInDatagrams/sec is the rate of UDP datagrams being delivered to UDP users.
	udpOutDatagrams/sec	udpOutDatagrams/sec is the rate at which UDP datagrams are sent.

QALoad-Linux Performance

This template includes the following counters and categories:

Category	Counters	Description
Linux System	CpuIdle%	CpuSys% is the percentage of idle processor time.
	CpuSys%	CpuSys% is the percentage of non-idle processor time that is spent in system mode.
	CpuUser%	CpuUser% is the percentage of non-idle processor time that is spent in user mode.
	Interrupts/sec	Interrupts/sec is the rate of system interrupts.
	PagesIn KBytes/sec	PagesIn KBytes/sec is the rate of pages read in from disk.
	PagesOut KBytes/sec	PagesOut KBytes/sec is the rate of pages written to disk.

	SwapIn KBytes/sec	SwapIn KBytes/sec is the rate at which pages are being swapped in.
	SwapOut KBytes/sec	SwapOut KBytes/sec is the rate at which pages are being swapped out.
tcp	tcpInSegs/sec	tcpInSegs/sec is the rate at which segments are received, including those received in error.
	tcpOutSegs/sec	tcpOutSegs/sec is the rate at which segments are sent, including those on current connections but excluding those containing only retransmitted octets.
udp	udpInDatagrams/sec	udpInDatagrams/sec is the rate of UDP datagrams being delivered to UDP users.
	udpOutDatagrams/sec	udpOutDatagrams/sec is the rate at which UDP datagrams are sent.

QALoad-SUN Performance

This template includes the following counters and categories:

Category	Counters	Description
Sun System	CpuIdle%	CpuSys% is the percentage of idle processor time.
	CpuSys%	CpuSys% is the percentage of non-idle processor time that is spent in system mode.
	CpuUser%	CpuUser% is the percentage of non-idle processor time that is spent in user mode.
	Interrupts/sec	Interrupts/sec is the rate of system interrupts.
	PagesIn KBytes/sec	PagesIn KBytes/sec is the rate of pages read in from disk.
	PagesOut KBytes/sec	PagesOut KBytes/sec is the rate of pages written to disk.
	SwapIn KBytes/sec	SwapIn KBytes/sec is the rate at which pages are being swapped in.
	SwapOut KBytes/sec	SwapOut KBytes/sec is the rate at which pages are being swapped out.
tcp	tcpInSegs/sec	tcpInSegs/sec is the rate at which segments are received, including those received in error.
	tcpOutSegs/sec	tcpOutSegs/sec is the rate at which segments are sent, including those on current connections but excluding those containing only retransmitted octets.

udp	udpInDatagrams/sec	udpInDatagrams/sec is the rate of UDP datagrams being delivered to UDP users.
	udpOutDatagrams/sec	udpOutDatagrams/sec is the rate at which UDP datagrams are sent.

WebLogic Templates

WebLogic Templates

QALoad provides the following pre-defined WebLogic templates:

QALoad-WebLogic Availability

QALoad-WebLogic EJB Performance

QALoad-WebLogic JDBC Performance

QALoad-WebLogic JMS Performance

QALoad-WebLogic Performance

QALoad-WebLogic Server Security

QALoad-WebLogic Servlet Performance

QALoad-WebLogic Availability

This template monitors the availability of a WebLogic server. The WebLogic Availability template returns critical information about the availability of your WebLogic installation.

The default sampling interval for this template is 5 minutes.

The WebLogic Availability template uses the following WebLogic extended counters:

Category	Counters	Description
ExecuteQueueRuntime	ExecuteQueueRuntime_PendingRequestOldestTime	Returns the time that the longest waiting request was placed in the queue. Rule: The Application Server is not in running mode if this counter value is > 50.
ServerRuntime	ServerRuntime_StateVal	Returns current state of the server. This counter provides a more detailed state than available or not. Rule: The Application Server is not in

		running mode if this counter value is <> 2.
ServerSecurityRuntime	ServerSecurityRuntime_LockedUsersCurrentCount	Returns the number of currently locked users on this server. Rule: There are a high number of users locked out if this counter value is > 5.

QALoad-WebLogic EJB Performance

This template monitors the EJB performance of a WebLogic server. The WebLogic EJB Performance template returns critical information about the performance of your WebLogic installation.

The default sampling interval for this template is 5 minutes.

The WebLogic EJB Performance template uses the following WebLogic extended counters:

Category	Counters	Description
EJBCacheRuntime	EJBCacheRuntime_ActivationCount	Returns the total number of times the EJB was activated. Rule: There is inefficient cache access if the number of activations is > 20.
	EJBCacheRuntime_CacheAccessCount	Returns the total number of attempts to access a bean from the cache.
	EJBCacheRuntime_CachedBeansCurrentCount	Returns the total number of beans from this EJB Home currently in the EJB cache.
	EJBCacheRuntime_CacheHitCount	Returns the total number of times an

		attempt to access a bean from the cache succeeded. The cacheHitCount value subtracting the cache miss count from the cache access count.
	EJBCacheRuntime_PassivationCount	Returns the total number of beans from this EJB Home that have been passivated. Rule: There is inefficient cache access if the number of passivations is > 20.
EJBLockingRuntime	EJBLockingRuntime_LockEntriesCurrentCount	Returns the number of currently locked users on this server.
	EJBLockingRuntime_TimeoutTotalCount	Returns the current number Threads that have timed out waiting for a lock on a bean.
	EJBLockingRuntime_WaiterTotalCount	Returns the number of objects waiting on the lock. Rule: There are a lot of objects waiting if the interval value

		of this counter is > 10.
EJBPoolRuntime	EJBPoolRuntime_BeansInUseCurrentCount	Returns the number of bean instances currently being used from the free pool.
	EJBPoolRuntime_IdleBeansCount	Returns the total number of available bean instances in the free pool.
	EJBPoolRuntime_TimeoutTotalCount	Returns the total number of Threads that have timed out waiting for an available bean instance from the free pool. Rule: There are a lot of objects timing out if the interval value of this counter is > 20.
	EJBPoolRuntime_WaiterTotalCount	Returns the total number of Threads currently waiting for an available bean instance from the free pool. Rule: There are a lot of objects waiting if the interval value of this counter is > 10.

EJBTransactionRuntime	EJBTransactionRuntime_TransactionsCommittedTotalCount	Returns the total number of transactions that have been committed for this EJB. Rule: There is high transaction overhead if the interval value of this counter is > 20.
	EJBTransactionRuntime_TransactionsRolledBackTotalCount	Returns the total number of transactions that have been rolled back for this EJB. Rule: There is high transaction overhead if the interval value of this counter is > 20.
	EJBTransactionRuntime_TransactionsTimedOutTotalCount	Returns the total number of transactions that have timed out for this EJB. Rule: There is high transaction overhead if the interval value of this counter is > 20.
MessageDrivenEJBRuntime	MessageDrivenEJBRuntime_JMSConnectionAlive	Returns a boolean of the status of

		the connection. This counter displays the state of a JMS connection. Rule: The JMS Connection is down if this counter value is = 0.

QALoad-WebLogic JDBC Performance

This template monitors the JDBC performance of a WebLogic server. The WebLogic JDBC Performance template returns critical information about the performance of your WebLogic installation.

The default sampling interval for this template is 5 minutes.

The WebLogic JDBC Performance template uses the following WebLogic extended counters:

Category	Counters	Description
JDBC Connection Pool Runtime	ActiveConnectionsCurrentCount	Returns the current number of active connections.
	ActiveConnectionsHighCount	Returns the highest number of active current connections. The count starts at zero each time the JDBCConnectionPoolRuntimeMBean is instantiated.
	ConnectionDelayTime	Returns the number of milliseconds it takes to get a physical connection from the database. It is calculated as summary time to connect divided by summary number of connections.
	ConnectionsTotalCount	Returns the total number of JDBC connections in this DBCConnectionPoolRuntimeMBean since the pool was instantiated.
	FailuresToReconnectCount	Returns the number of attempts to refresh a connection to a database that failed. Failure may be due to the database being unavailable or a broken connection to the database. Rule: There are a high number of connection reconnect failures when this counter value is > 1.
	LeakedConnectionCount	Returns the number of connections that were checked out from the connection pool but were not returned to the pool by calling close

		(). Rule: There is a lot of connection pool leakage if this counter value is > 5.
	PoolState	Current state of the connection pool. Returns True if the pool is enabled, False if the pool is disabled.
	PrepStmtCacheMissCount	Returns a count of the cases when the cache does not have a cached statement to satisfy a request.
	WaitingForConnectionHighCount	The high water mark of waiters for a connection in this JDBCConnectionPoolRuntimeMBean. The count starts at zero each time the JDBCConnectionPoolRuntimeMBean is instantiated.
	WaitSecondsHighCount	Returns the highest number of seconds a connection waited. Rule: There is a long wait for the connection pool if this counter value is > 120.

QALoad-WebLogic JMS Performance

This template monitors the JMS performance of a WebLogic server. The WebLogic JMS Performance template returns critical information about the performance of your WebLogic installation.

The default sampling interval for this template is 5 minutes.

The WebLogic JMS Performance template uses the following WebLogic extended counters:

Category	Counters	Description
JMSConnectionRuntime	SessionsCurrentCount	Returns the current number of sessions for this connection.
	SessionsTotalCount	Returns the number of sessions on this connection since the last reset.
JMSRuntime	ConnectionsCurrentCount	Returns the current number of connections to this WebLogic Server.
	ConnectionsTotalCount	Returns the total number of connections made to this WebLogic Server since the last reset.
JMSServerRuntime	MessagesPendingCount	Returns the current number of messages pending (unacknowledged or uncommitted) stored on this

		JMS server. Pending messages are over and above the current number of messages. Rule: There are a large number of pending messages if this counter value is > 50.
	MessagesReceivedCount	Returns the number of messages received on this destination since the last reset.
JMSSessionRuntime	ConsumersCurrentCount	Returns the current number of consumers for this session.
	MessagesPendingCount	Returns the number of messages pending (uncommitted and unacknowledged) for this session. Rule: There are a large number of pending JMS Session messages if this counter value is > 50.
	MessagesReceivedCount	Returns the number of messages received on this destination since the last reset.
	MessagesSentCount	Returns the number of bytes sent by this session since the last reset.

QALoad-WebLogic Performance

This template monitors the performance of a WebLogic server. The WebLogic Performance template returns critical information about the performance of your WebLogic installation.

The default sampling interval for this template is 5 minutes.

The WebLogic Performance template uses the following WebLogic extended counters:

Category	Counters	Description
ConnectorServiceRuntime	ConnectionPoolCurrentCount	Returns the number of currently deployed connection pools.
ExecuteQueueRuntime	ExecuteThreadCurrentIdleCount	Returns the number of idle threads assigned to the queue.
	PendingRequestCurrentCount	Returns the number of waiting requests in the queue. Rule: There are a large number of pending requests if this counter value is > 50.

	ServicedRequestTotalCount	Returns the number of requests that have been processed by this queue.
JMSRuntime	ConnectionsCurrentCount	Returns the current number of connections to this WebLogic Server. Rule: There are a large number of JMS connections if this counter value is > 20.
JTARuntime	ActiveTransactionsTotalCount	Returns the number of active transactions on the server.
	SecondsActiveTotalCount	Returns the total number of seconds for all committed transactions.
	TransactionRolledBackResourceTotalCount	Returns the number of transactions that were rolled back due to a resource error.
	TransactionTotalCount	Returns the total number of transactions processed. This total includes all committed, rolled back and heuristic transaction completions.
JVMRuntime	HeapFreeCurrent	Returns the current amount of free memory (in bytes) in the JVM heap.
TimeServiceRuntime	ExceptionCount	Returns the total number of exceptions thrown while executing scheduled triggers. Rule: There are a large number of exceptions if the interval value of this counter is > 20.
	ExecutionsPerMinute	Returns the average number of triggers executed per minute.

QALoad-WebLogic Server Security

This template monitors the security of a WebLogic server. The WebLogic Server Security template returns critical information about the security status of your WebLogic installation.

The default sampling interval for this template is 5 minutes.

The WebLogic Server Security template uses the following WebLogic extended counters:

Category	Counters	Description

ServerSecurityRuntime	InvalidLoginAttemptsTotalCount	Returns the cumulative number of invalid login attempts made on this server. Rule: Multiple invalid login attempts have occurred when the interval value of this counter is > 5.
	LockedUsersCurrentCount	Returns the number of currently locked users on this server. Rule: There are multiple locked users if this counter value is > 5.
	LoginAttemptsWhileLockedTotalCount	Returns the cumulative number of invalid login attempts made on this server while the user was locked.
	UnlockedUsersTotalCount	Returns the number times users have been unlocked on this server.

QALoad-WebLogic Servlet Performance

This template monitors the performance of your WebLogic servlet. The WebLogic Servlet Performance template returns critical information about the servlet performance of your WebLogic installation.

The default sampling interval for this template is 5 minutes.

The WebLogic Servlet Performance template uses the following WebLogic extended counters:

Category	Counters	Description
ServletRuntime	ExecutionTimeAverage	Returns the average time all invocations of the servlet that has executed since the task was created. Rule: The servlet is averaging high execution times if this counter value average is > 10.
	ExecutionTimeHigh	Returns the amount of time the single longest invocation of the servlet that has executed since the task was created.
	ExecutionTimeTotal	Returns the total amount of time all invocations of the servlet that has executed since the task was created.
	InternalServlet	whether this is an Internal Servlet or not
	InvocationTotalCount	Returns the total number of times the servlet has been invoked. Gets the invocationTotalCount attribute of the ServletRuntimeMBean object.
	ReloadTotalCount	Returns the total number of times the servlet is reloaded. Gets the reloadTotalCount attribute of the ServletRuntimeMBean object.

WebSphere Templates

WebSphere Templates

QALoad provides the following pre-defined WebSphere templates:

QALoad-WebSphere 5.0 JDBC Performance

QALoad WebSphere 5.0 Performance

QALoad-WebSphere 5.0 Web Application Performance

QALoad-WebSphere 5.0 JDBC Performance

This template monitors the performance of a WebSphere JDBC server. The WebSphere JDBC Performance template returns critical information about the JDBC performance of your WebSphere installation.

The default sampling interval for this template is 5 minutes.

The WebSphere JDBC Performance template uses the following WebSphere extended counters:

Category	Counters	Description
JDBC Connection Pool Module	connectionPoolModule.avgWaitTime	Average waiting time in milliseconds until a connection
	connectionPoolModule.concurrentWaiters	WebSphere extended counter for monitoring connectionPoolModule.concurrentWaiters
	connectionPoolModule.faults	Average waiting time in milliseconds until a connection
	connectionPoolModule.percentMaxed	Average percent of the time that all connections are in u Rule: IF 'WebSphere connectionPoolModule.connectionPoolModule.percentM "**", Server: "**", Data Source: "all")' > 25.
	connectionPoolModule.percentUsed	Average percent of the pool that is in use.

QALoad-WebSphere 5.0 Web Application Performance

This template monitors the performance of a WebSphere 5.0 Web Application server. The WebSphere 5.0 Web Application Performance template returns critical information about the Web Application performance of your WebSphere installation.

The default sampling interval for this template is 5 minutes.

The WebSphere 5.0 Web Application Performance template uses the following WebSphere extended counters:

Category	Counters	Description
WebSphere servletSessionsModule	servletSessionsModule.activateNonExistSessions	Number of requests for a session that no longer exists, presumably because the session timed out. This counter may indicate a high number of timeout conditions.

	servletSessionsModule.activeSessions	The number of concurrently active sessions. A session is active if WebSphere is currently processing a request, which uses that session. This counter may indicate high activity.
	servletSessionsModule.cacheDiscards	Number of session objects that have been forced out of the cache. This counter may indicate a need for more memory in the cache.
	servletSessionsModule.invalidatedSessions	Number of sessions invalidated. This counter may indicate a high number of invalidated sessions.
	servletSessionsModule.invalidatedViaTimeout	Number of requests for a session that no CountStatistic exists, presumably because the session timed out. This counter may indicate a high number of timeout conditions.
WebSphere webAppModule	webAppModule.servlets.concurrentRequests	Number of requests that are concurrently processed. This counter may indicate high activity for an application.
	webAppModule.servlets.numErrors	Total number of errors in a servlet or Java Server Page (JSP). This counter may indicate a high number of error incidents.
	webAppModule.servlets.responseTime	Response time, in milliseconds, of a servlet request. This counter may indicate a slow response time of a request.

QALoad WebSphere 5.0 Performance

This template includes the following counters and categories:

Category	Counters	Description
WebSphere jvmRuntimeModule	jvmRuntimeModule.freeMemory	WebSphere extended counter for monitoring jvmRuntimeModule.freeMemory

	jvmRuntimeModule.usedMemory	WebSphere extended counter for monitoring jvmRuntimeModule.usedMemory
WebSphere orbPerfModule	orbPerfModule.concurrentRequests	WebSphere extended counter for monitoring orbPerfModule.concurrentRequests
	orbPerfModule.interceptors.processingTime	WebSphere extended counter for monitoring orbPerfModule.interceptors.processingTime
	orbPerfModule.referenceLookupTime	WebSphere extended counter for monitoring orbPerfModule.referenceLookupTime
WebSphere systemModule	systemModule.avgCpuUtilization	WebSphere extended counter for monitoring systemModule.avgCpuUtilization
	systemModule.freeMemory	WebSphere extended counter for monitoring systemModule.freeMemory
WebSphere threadPoolModule	hreadPoolModule.activeThreads	WebSphere extended counter for monitoring threadPoolModule.activeThreads

WebSphere MQ Templates

WebSphere MQ Templates

QALoad provides the following pre-defined WebSphere MQ templates:

QALoad-WebSphere MQ Availability

QALoad-WebSphere MQ Performance

QALoad-WebSphere MQ Availability

This template monitors the availability of a WebSphere MQ server. The WebSphere MQ Availability template returns critical information about the availability of your WebSphere MQ installation.

The default sampling interval for this template is 5 minutes.

The WebSphere MQ Availability template uses the following WebSphere MQ extended counters:

Counters	Description
Channel Events	Return the number of channel events for the current interval.
Queue Manager Events	Reports the number of queue manager events for the current interval.
Queue Manager Up/Down	Monitors the running state of a queue manager.

QALoad-WebSphere MQ Performance

This template monitors the performance of a WebSphere MQ server. The WebSphere MQ Performance template returns critical information about the performance of your WebSphere MQ installation.

The default sampling interval for this template is 5 minutes.

The WebSphere MQ Performance template uses the following WebSphere MQ extended counters:

Counters	Description
Performance Events	This counter reports the number of performance events for the current interval.

WMI Templates

WMI Templates

QALoad provides the following pre-defined WMI templates:

QALoad-Active Monitoring Availability

QALoad-Citrix IMA Networking

QALoad-Citrix Metaframe All

QALoad-Citrix MetaFrame IMA

QALoad-Citrix MetaFrame Zone

QALoad-Cold Fusion

QALoad-Generic Application Availability and Performance

QALoad-MS IIS Availability

QALoad-MS IIS Performance

QALoad-Active Monitoring Availability

This template includes the following counters and categories:

Category	Counters	Description
Memory	Available MBytes	
Processor	% Processor Time	
System	System Up Time	

QALoad-Citrix IMA Networking

This template includes the following counters and categories:

Category	Counters	Description
Citrix IMA Networking	Bytes Received/sec	
	Bytes Sent/sec	
	Network Connections	

Network Interface	Bytes Total/sec	

QALoad-Citrix Metaframe All

This template includes the following counters and categories:

Category	Counters	Description
Citrix MetaFrame XP	Application Enumerations/sec	
	Application Resolution Time (ms)	
	Application Resolutions/sec	
	Data Store Connection Failure	
	DataStore bytes read/sec	
	DataStore bytes written/sec	
	DataStore reads/sec	
	DataStore writes/sec	
	Dynamic Store bytes read/sec	
	DynamicStore bytes written/sec	
	DynamicStore reads/sec	
	DynamicStore writes/sec	
	Filtered Application Enumerations/sec	
	LocalHostCache bytes read/sec	
	LocalHostCache bytes written/sec	
	LocalHostCache reads/sec	
	LocalHostCache writes/sec	
	Zone Elections	
	Zone Elections Won	
Memory	Page Reads/sec	
PhysicalDisk	% Disk Time	
Processor	% Processor Time	

QALoad-Citrix MetaFrame IMA

This template includes the following counters and categories:

Category	Counters	Description
Citrix MetaFrame XP	Application Enumerations/sec	
	Application Resolution Time (ms)	
	Application Resolutions/sec	
	Data Store Connection Failure	
	DataStore bytes read/sec	
	DataStore bytes written/sec	
	DataStore reads/sec	
	DataStore writes/sec	
	Filtered Application Enumerations/sec	
	LocalHostCache bytes read/sec	
	LocalHostCache bytes written/sec	
	LocalHostCache reads/sec	
	LocalHostCache writes/sec	
Terminal Services	Active Sessions	
	Total Sessions	

QALoad-Citrix MetaFrame Zone

This template includes the following counters and categories:

Category	Counters	Description
Citrix MetaFrame XP	Dynamic Store bytes read/sec	
	DynamicStore bytes written/sec	
	DynamicStore reads/sec	
	DynamicStore writes/sec	
	LocalHostCache bytes read/sec	
	LocalHostCache bytes written/sec	

	LocalHostCache reads/sec	
	Zone Elections	
	Zone Elections Won	
Network Interface	Bytes Total/sec	
	Current Bandwidth	
Terminal Services	Active Sessions	
	Total Sessions	

QALoad-Cold Fusion

This template includes the following counters and categories:

Category	Counters	Description
ColdFusion MX Server	Avg DB Time (msec)	
	Avg Queue Time (msec)	
	Avg Req Time (msec)	
	Bytes In / Sec	
	Bytes Out / Sec	
	DB Hits / Sec	
	Page Hits / Sec	
	Queued Requests	
	Running Requests	
	Timed Out Requests	
Memory	% Committed Bytes In Use	
	Available Bytes	
	Page Faults/sec	
Process	% Processor Time	

QALoad-Generic Application Availability and Performance

This template includes the following counters and categories:

Category	Counters	Description
Process	% Processor Time	

System	System Up Time	

QALoad-MS IIS Availability

This template includes the following counters and categories:

Category	Counters	Description
System	System Up Time	
Web Service	Current Anonymous Users	
	Current Connections	
	Logon Attempts/sec	
	NonAnonymous Users/sec	
	Not Found Errors/sec	
	Total Delete Requests	
	Total Files Sent	
	Total Get Requests	
	Total NonAnonymous Users	
	Total Not Found Errors	

QALoad-MS IIS Performance

This template includes the following counters and categories:

Category	Counters	Description
Internet Information Services Global	Current Blocked Async I/O Requests	
	Total Blocked Async I/O Requests	
	Total Rejected Async I/O Requests	
	URI Cache Flushes	
	URI Cache Hits	
	URI Cache Hits %	
	URI Cache Misses	
PhysicalDisk	% Disk Time	

Process	% Processor Time	
Redirector	Current Commands	
	Network Errors/sec	
Server	Work Item Shortages	
Server Work Queues	Queue Length	
Web Service	Not Found Errors/sec	

Windows Registry Templates

Windows Registry Templates

QALoad provides the following pre-defined Windows Registry templates:

QALoad-Active Monitoring Availability

QALoad-Citrix IMA Networking

QALoad-Citrix Metaframe all

QALoad-Citrix Metaframe IMA

QALoad-Citrix Metaframe Zone

QALoad-Cold Fusion

QALoad-Generic Application Availability and Performance

QALoad-MS IIS Availability

QALoad-MS IIS Performance

QALoad-Server Health

QALoad-Windows Availability

QALoad-Windows Performance

QALoad-Active Monitoring Availability

This template includes the following counters and categories:

Category	Counters	Description
Memory	Available MBytes	This counter monitors the Active Monitoring client site and notifies you when it is low on resources, where Processor time is > 95% for more than 3 intervals. The parameter for this counter is Instance. The default is _Total.
Processor	% Processor Time	Raise an event when Active Monitoring client site is low on memory resources, where Available Memory is at or below 1MB for more than 3 intervals.

System	System Up Time	This counter tests the network connection between two machines and monitors the communication status of the machine that receives communication.

QALoad-Citrix IMA Networking

This template includes the following counters and categories:

Category	Counters	Description
Citrix IMA Networking	Bytes Received/sec("_Total")	This counter monitors the total bytes received per second.
	Bytes Sent/sec("_Total")	This counter monitors the total bytes sent per second.
	Network Connections	This counter monitors the network connections.
Network Interface	Bytes Total/sec	This counter monitors the network connection total bytes/sec.

QALoad-Citrix Metaframe all

This template includes the following counters and categories:

Category	Counters	Description
Citrix MetaFrame XP	Application Enumerations/sec	This counter monitors application enumerations / sec.
	Application Resolution Time (ms)	This counter monitors application resolution time.
	Application Resolutions/sec	This counter monitors application resolutions.
	Data Store Connection Failure	This counter monitors datastore connection failure.
	DataStore bytes read/sec	This counter monitors datastore bytes reads per second.
	DataStore bytes written/sec	This counter monitors datastore bytes written per second.
	DataStore reads/sec	This counter monitors datastore reads per second.

	DataStore writes/sec	This counter monitors datastore writes per second.
	Dynamic Store bytes read/sec	This counter monitors DynamicStore bytes read per second.
	DynamicStore bytes written/sec	This counter monitors DynamicStore bytes written per second.
	DynamicStore reads/sec	This counter monitors DynamicStore reads per second.
	DynamicStore writes/sec	This counter monitors DynamicStore writes per second.
	Filtered Application Enumerations/sec	This counter monitors Filtered Application Enumerations per second.
	LocalHostCache bytes read/sec	This counter monitors LoadHostCache bytes read per second.
	LocalHostCache bytes written/sec	This counter monitors LoadHostCache bytes written per second.
	LocalHostCache reads/sec	This counter monitors LoadHostCache reads per second.
	LocalHostCache writes/sec	This counter monitors LoadHostCache writes per second.
	Zone Elections	This counter monitors zone elections.
	Zone Elections Won	This counter monitors zone elections won.
Memory	Page Reads/sec	This counter monitors page reads per second.
PhysicalDisk	% Disk Time	This counter monitors % disk time.
Processor	% Processor Time	This counter monitors % processor time.

QALoad-Citrix Metaframe IMA

This template includes the following counters and categories:

Category	Counters	Description
Citrix MetaFrame XP	Application Enumerations/sec	This counter monitors the application enumeration per second.
	Application Resolution Time (ms)	This counter monitors the application resolution time.
	Application Resolutions/sec	This counter monitors the application resolution.
	Data Store Connection Failure	This counter monitors the datastore connection failure.
	DataStore bytes read/sec	This counter monitors the datastore bytes read per second.
	DataStore bytes written/sec	This counter monitors the datastore bytes written per second.
	DataStore reads/sec	This counter monitors the datastore reads per second.
	DataStore writes/sec	This counter monitors the datastore writes per second.
	Filtered Application Enumerations/sec	This counter monitors filtered application enumerations per second.
	LocalHostCache bytes read/sec	This counter monitors LoadHostCache bytes read per second.
	LocalHostCache bytes written/sec	This counter monitors LoadHostCache bytes written per second.
	LocalHostCache reads/sec	This counter monitors LoadHostCache reads per second.
	LocalHostCache writes/sec	This counter monitors LoadHostCache writes per second.
Terminal Services	Active Sessions	This counter monitors active sessions.
	Total Sessions	This counter monitors total sessions.

QALoad-Citrix Metaframe Zone

This template includes the following counters and categories:

Category	Counters	Description
Citrix MetaFrame XP	Dynamic Store bytes read/sec	This counter monitors the dynamic store bytes read / sec.
	DynamicStore bytes written/sec	This counter monitors the dynamic store bytes written / sec.
	DynamicStore reads/sec	This counter monitors the dynamic store reads / sec.
	DynamicStore writes/sec	This counter monitors the dynamic store writes / sec.
	LocalHostCache bytes read/sec	This counter monitors the LocalHostCache bytes read / sec.
	LocalHostCache bytes written/sec	This counter monitors the LocalHostCache bytes written / sec.
	LocalHostCache reads/sec	This counter monitors the LocalHostCache reads / sec.
	Zone Elections	This counter monitors the zone elections.
	Zone Elections Won	This counter monitors the zone elections won.
Network Interface	Bytes Total/sec	This counter monitors network connection total bytes.
	Current Bandwidth	This counter monitors network connection current bandwidth.
Terminal Services	Active Sessions	This counter monitors active sessions.
	Total Sessions	This counter monitors total sessions.

QALoad-Cold Fusion

This template includes the following counters and categories:

Category	Counters	Description
ColdFusion MX Server	Avg DB Time (msec)	
	Avg Queue Time (msec)	

	Avg Req Time (msec)	
	Bytes In / Sec	
	Bytes Out / Sec	
	DB Hits / Sec	
	Page Hits / Sec	
	Queued Requests	
	Running Requests	
	Timed Out Requests	
Memory	% Committed Bytes In Use	
	Available Bytes	
	Page Faults/sec	
Process	% Processor Time	

QALoad-Generic Application Availability and Performance

This template includes the following counters and categories:

Category	Counters	Description
Process	% Processor Time	This counter returns the percentage of elapsed time that all threads of a process use the processor to execute instructions. This process could include code executed to handle certain hardware interrupts or trap conditions.
System	System Up Time	This counter monitors critical tasks by verifying the existence of processes. You can monitor single or multiple tasks running on the system by selecting the Processes tab from the task manager and then selecting processes that you want to monitor. You can also monitor only certain instances of a task by specifying a Process ID to monitor. If you do not specify a Process ID, this counter monitors all instances of the task.

QALoad-MS IIS Availability

This template includes the following counters and categories:

Category	Counters	Description
System	System Up Time	
Web Service	Current Anonymous Users	
	Current Connections	
	Logon Attempts/sec	
	NonAnonymous Users/sec	
	Not Found Errors/sec	
	Total Delete Requests	
	Total Files Sent	
	Total Get Requests	
	Total NonAnonymous Users	
	Total Not Found Errors	

QALoad-MS IIS Performance

This template includes the following counters and categories:

Category	Counters	Description
Internet Information Services Global	Current Blocked Async I/O Requests	
	Total Blocked Async I/O Requests	
	Total Rejected Async I/O Requests	
	URI Cache Flushes	
	URI Cache Hits	
	URI Cache Hits %	
	URI Cache Misses	
PhysicalDisk	% Disk Time	
Process	% Processor Time	
Redirector	Current Commands	

	Network Errors/sec	
Server	Work Item Shortages	
Server Work Queues	Queue Length	
Web Service	Not Found Errors/sec	

QALoad-Server Health

This template includes the following counters and categories:

Category	Counters	Description
Memory	% Committed Bytes In Use	
	Pages/sec	
PhysicalDisk	% Disk Time	
	Avg. Disk Queue Length	
Processor	% Processor Time	
System	Processor Queue Length	

QALoad-Windows Availability

This template monitors the availability of the Windows operating system, focusing on:

Logons

Security

Up time

The default sampling interval for this template is 5 minutes.

This template includes the following counters and categories:

Category	Counters	Description
Server	Errors Access Permissions Errors Granted Access Errors Logon Errors System Logon Total	The Microsoft Windows Availability template uses these Server registry counters to monitor errors due to logon problems. To enable these counters, you must configure your Windows system to audit logon and logoff events. You can do this by configuring the Audit Policy in the User Manager for Domains program.
	Server Sessions	The Microsoft Windows Availability template uses these

	Sessions Errored Out Sessions Forced Off Sessions Logged Off Sessions Timed Out	Server registry counters to monitor how well users' sessions are running. If there is a large number of session errors, it is usually due to systems rebooting often or network errors.
System	System Up Time	This counter returns the number of seconds that a system was available for use. If this number continues to reset to zero, it means that the system is rebooting often. For a report that lists the number of times that the system has rebooted over a period of time, see the Microsoft Windows Availability Report topic.

QALoad-Windows Performance

This template monitors the performance of the Microsoft Windows system, focusing on:

CPU

Disk I/O

Disk space

Memory

Network

The default sampling interval for this template is 5 minutes.

This template includes the following counters and categories:

Category	Counters	Description
LogicalDisk	% Disk Time	This counter monitors the percentage of elapsed time that the disk services read and write requests, including the time that the disk driver waits in the disk queue. If this value is consistently near 100%, the disk is in very heavy use. You can determine which processes are making the majority of the disk requests by monitoring them individually.
	% Free Space	This counter monitors low free-space situations.
	Avg. Disk Queue Length	This counter indicates the number of pending I/O service requests. If the returned value is

		greater than 2, there is a disk problem. On a multi-disk subsystem, such as a striped set or striped with parity, you can perform a calculation to determine the presence of a disk problem. The basic formula is (Disk Queue Length) - (Number of Physical Disk Drives in the multi-disk configuration). For example, if you have a striped set with 3 disk drives and a queue length of 5, then you get an acceptable value of 2 (5 - 3 = 2).
Memory	Available Bytes	If the value returned by this counter falls under 10 MB, virtual memory is running low. To resolve this, close some applications or increase the memory settings. If this counter is consistently low after an application is running, it usually indicates a system memory leak. As the value returned by this counter decreases, the value returned by the Committed Bytes counter increases. This indicates that a process is allocating memory from the virtual address space but might not be using it. Because the virtual address space is a limited resource, use these counters to check for applications that allocate memory but do not use it. To resolve this, add more physical memory. When an application finishes processing, note the last value. If this counter does not return to the original value, the application has a memory leak or a hidden process that has not properly terminated. The acceptable range for committed bytes should be less than the physical RAM. The default value is 64 MB.
	Cache Faults/sec	If the value returned by this counter is less than the value returned by the Page faults/sec

		counter, the system is paging too much for a normal system. To resolve this, add more physical memory.
	Committed Bytes	This counter returns the amount of virtual memory (in bytes) that was committed, as opposed to memory that has was reserved.
	Page Faults/sec	If the value returned by this counter is greater than 5, the system is paging too much. Add more physical memory. A consistent value of 10 or later needs immediate attention.
	Page Reads/sec	
	Pages/sec	If this counter returns a high peak value, the system is experiencing a lot of paging activity. A high value also indicates that your system does not contain enough physical memory to handle the demands placed on it by the application. To resolve this, add more physical memory. To calculate the % disk time used for paging, use the following calculation: (% Disk Time used for paging) = (Memory, Pages/sec) * (Average Disk Transfer/sec) * 100
	Transition Faults/sec	
Paging File	% Usage Peak	This counter returns the maximum use of your page file. If the value the counter returns consistently reaches 90%, the virtual address space is too small. You should increase the size of your paging file. When the value returned by the counter exceeds 75%, a significant system performance degradation becomes noticeable.
PhysicalDisk	% Disk Time	
	Avg. Disk Queue Length	
	Avg. Disk sec/Transfer	

	Disk Reads/sec	
	Disk Writes/sec	
Processor	% Interrupt Time	This counter monitors the percentage of time that the processor spent receiving and servicing hardware interrupts during the sample interval.
	% Processor Time	On single processor systems, if the value returned by this counter is consistently higher than 90%, the CPU probably has a bottleneck. You should examine each process in the system to determine which one is using more of the processor than it should. The process with the highest peak is generally the performance bottleneck.
	% User Time	This counter monitors non-idle processor time spent in User mode as a percentage of the sample interval.
Redirector	Network Errors/sec	This counter indicates how many serious network errors have occurred. These errors are generally logged in the system event log, so you can check there for more information. If an error occurs, take immediate action to resolve the problem.
Server	Bytes Received/sec	
	Bytes Total/sec	
	Bytes Transmitted/sec	
	Errors Logon	This counter determines if an unauthorized user is trying to access your system.
	Work Item Shortages	This counter monitors the number of times that a work item was not allocated. You might need to increase the InitWorkItems and MaxWorkItems parameters for the LanMan Server if this number continues to increase.
System	Context Switches/sec	If the value returned by this

		counter value is high, assign a higher priority to the use of critical sections or semaphores by the program. This achieves a higher throughput and reduces task switching.
	Processor Queue Length	

Managing Monitoring Templates

Creating a New Template

To open the New Monitoring Template wizard:

1. In Conductor, click Tools>Monitor Tasks to open the Manage Monitoring Tasks window.
2. Click Templates>New Template. The New Monitoring Template Wizard appears.
3. Click Next to start the procedure.

To create a new template:

Use the following steps in the New Monitoring Template wizard to create a new monitoring template:

Note: (WebLogic and WebSphere) When QALoad detects a managed server environment, you must also select the individual server on which to model the template.

1. Enter the template properties
2. Configure the monitor
3. Server discovery (WebLogic and WebSphere)
4. Choose the server (WebLogic and WebSphere)
5. Process the server (WebLogic and WebSphere)
6. Counter discovery
7. Choose the counters
8. Choose the instances
9. Review, save, and create the template

Opening an Existing Template

Use the following steps to apply a previously created or pre-defined template.

To open and review an existing template:

1. In the Conductor, click Tools>Monitor Tasks to open the Manage Monitoring Tasks window.
2. In the Manage Monitoring Tasks window, click Templates>Open Existing. The Select a Monitor Template File dialog box displays.
3. In the Look in field, select a template type, then select a template and click Open. The template and its counters display in the Manage Monitoring Tasks window.

Note: To apply a template to a task, use the New Monitoring Task wizard.

Editing Instances for Templates

Use the following steps to modify the instances to monitor in a custom template:

Step 1: Open the Edit Template Instances Wizard

Step 2: Choose the instances to monitor

Step 3: Save the template

Step 1: Open the Edit Template Instances wizard:

1. In the Conductor, click Tools>Monitor Tasks to display the Manage Monitoring Tasks window.
2. Open the template to edit.
3. Click Templates>Edit instances. The Edit Template Instances Wizard appears.

Step 2: Choose the instances of the counter to monitor:

Review the counters you selected. When a red dot appears next to a counter, you must select an instance for the counter.

1. Double-click the counter group to display the counters.
2. Select an instance for a counter and click Edit. The Select instance for counter dialog box appears.
3. To add an instance: In the Available Instance pane, select an instance and click Add.
4. To remove an instance: In the Selected instances pane, select an instance and click Remove.
5. Repeat until you select all instances of the counter that you want to apply to the task.
6. Click Save. You return to the Choose Instances dialog box.
7. Repeat this process for each designated counter.
8. Click Next. The Summary dialog box displays.

Step 3: Save the template:

1. On the Summary dialog box, review the monitors and counters you have selected for the template. Click Back to return to a dialog box and make changes to the information.
2. Click Back to return to the previous step and edit the instances.
3. Click Finish to create the template.

Modifying Template Counters Using New Discovery Data

When you want to add or edit counters in a custom template, you can generate the discovery data that you add to the template. The Edit Monitoring Template wizard guides you through the process of generating and applying new discovery data. Use the following steps to modify template counters using new discovery:

Step 1: Open the Edit Monitoring Template Wizard

Step 2: Enter properties of the template

Step 3: Configure the monitor

Step 4: Counter Discovery

Step 5: Choose the counters

Step 6: Choose the instances of the counter

Step 7: Save the template

Step 1: Open the Edit Monitoring Template wizard:

1. In the Manage Monitoring Tasks window, open the template to edit.
2. Click Templates>Add/Edit Counter>Use new discovery data. The Edit Monitoring Template wizard appears.
3. Click Next in the Welcome dialog box. The Enter properties of the template dialog box displays.

Note: (WebLogic and WebSphere) You can change the Java Settings field, if necessary. If the Java file location has changed, click the Browse button and select the new location.

Step 2: Enter properties of the template:

In the Enter properties of the template dialog box, do the following:

1. Review the template information.
2. Type or edit the description for the template in the Description field
3. Click Next. The Configure Monitor dialog box displays.

Note: (WebLogic and WebSphere) You can modify any field in the Configure Monitor dialog box except the Admin Server field.

Step 3: Configure the monitor:

In the Configure Monitor Dialog, do the following:

1. Type the configuration data for the host machine, if necessary. This data is used to connect to the host machine and to the host database during counter discovery and runtime data collection. The required configuration data varies depending on the monitor type selected. Click a link below to view the required configuration details for your monitor type.
   - Oracle Application Server
   - JVM
   - SAP
   - ServerVantage
   - SNMP
   - WebLogic
   - WebSphere
   - WebSphere MQ
   - Windows Registry
   - WMI
2. Click Next.

- WebLogic and WebSphere) The Processing Servers dialog box displays. Follow the procedure for selecting a server. Once you select the server, the automatic counter discovery process begins.
- (All other monitor types) The automatic counter discovery process begins.

Step 4: Counter discovery:

QALoad automatically performs the counter discovery. The default maximum time for counter discovery is 300 seconds. When counter discovery is complete, the Choose Counters dialog box displays.

Step 5: Choose the counters:

When the counter discovery process completes, the Add the desired counter to this template dialog box appears.

1. From the Available Items pane in the Choose Counters dialog box, select the Template tab or the Counter tab.
2. To add an item, select a template or a counter to monitor in the task for this machine and monitor type, and click Add, or double-click the item to display it in the Selected Items pane. Click Add All to add all the items on the selected tab to the Selected Items pane.
3. To remove an item, double-click the item in the Selected Items pane or select the item and click Remove. The item is returned to the Available Items pane.

Note: Select multiple counters and templates by doing one of the following:

- To select nonadjacent counter items, click one counter item, and then hold down CTRL and click each additional counter item.
- To select adjacent counter items, click the first counter item in the sequence, and then hold down SHIFT and click the last counter item.

4. Click Next. The Choose Instances dialog box displays.

Note: When you select a template containing counters that are not present on the machine you are defining, you receive a message with a list of the counters that will not be added to the task.

Step 6: Choose the instances of the counter to monitor:

Review the selected counters. When a red dot appears next to a counter, select an instance of the counter.

1. Double-click the counter group to display the counters.
2. Select a counter and click Edit. The Select instance for counter dialog box appears.
3. In the Available Instance pane, select an instance and click Add.
4. Repeat until you select all instances of the counter that you want to apply to the task.
5. Click Save. The Choose Instances dialog box appears.
6. Repeat this process for each designated counter.
7. Click Next. The Summary dialog box displays.

Step 7: Save the template:

1. On the Summary dialog box, review the counters and instances you have selected for the template. Click Back to return to a dialog box and make changes to the information.

2. Click Finish to create the template.

Modifying Template Counters Using Cached Discovery

When you need to add or edit counters in a template that you created, you can use the cached counter discovery data to modify the template.

Note: You cannot modify the counters in pre-defined templates.

Follow these steps to modify template counters using cached discovery:

Step 1: Select the counter to add or remove

Step 2: Choose the instances of the counter

Step 3: Save the template

Step 1: Select the counter to add or remove:

1. In Conductor, click Tools>Monitor Tasks to open the Manage Monitoring Tasks window.
2. Open the template to edit, then click Templates>Add/Edit counter>Use cached discovery data. The Edit Template Counters wizard appears with the Add/Edit/Remove Template Counters dialog box displayed.
3. From the Available Items pane, select the Template tab or the Counter tab.
4. To add an item, select a template or a counter to monitor for this machine and monitor type, and click Add, or double-click the item to display it in the Selected Items pane. Click Add All to add all the items on the selected tab to the Selected Items pane.
5. To remove an item, select the item in the Selected Items pane and click Remove, or double-click the item to return it to the Available Items pane.

Note: Select multiple counters and templates by doing one of the following:

! To select nonadjacent counter items, click one counter item, and then hold down Ctrl and click each additional counter item.

! To select adjacent counter items, click the first counter item in the sequence, and then hold down Shift and click the last counter item.

6. Click Next. The Add/Edit/Remove Template Instances dialog box displays.

Note: When you select a template that contains counters not present on the machine you are defining, a message displays with a list of the counters that will not be added.

Step 2: Choose the instances of the counter to monitor:

1. Review the selected counters. When a red dot appears next to a counter, select an instance of the counter.
2. Double-click the counter group to display the counters.
3. Select a counter and click Edit. The Select instance for counter dialog box appears.
4. In the Available Instance pane, select an instance and click Add.
5. Repeat until you select all instances of the counter that you want to apply.
6. Click Save. The Choose Instances dialog box appears.
7. Repeat this process for each designated counter.

8. Click Next. The Summary dialog box displays.

Step 3: Save the template:

1. On the Summary dialog box, review the selected monitors and counters for the template. Click Back to return to a dialog box and make changes to the information.
2. Click Finish to create the template.

Removing a Counter from a Template

Remove a counter from a template by following this procedure.

To remove a counter from a template:

1. In Conductor, click Tools>Monitor Tasks to open the Manage Monitoring Tasks window.
2. Select the counter or counter family to delete.
3. Click Templates>Remove counter.
4. When the verification dialog box displays, click OK.

Note: You cannot remove the only counter family in a template or the last counter in a family.

Creating and Editing Monitoring Tasks

Creating a New Monitoring Task

To open the New Monitoring Task wizard:

1. From the Conductor Start Page, click Configure Monitoring in the Tasks area.

   OR

   In the Conductor's Visual Designer, click Tools>Monitor Tasks to open the Manage Monitoring Tasks window.
2. Click File>New. The New Monitoring Task wizard appears.
3. Click Next to start the procedure.

Note: You can open the Manage Monitoring Tasks window to edit or create a task by clicking the browse button next to the Monitor task field in the Session node.

To create a new monitoring task:

Use the following steps in the New Monitoring Task Wizard to create a new monitoring task:

1. Define the monitor
2. Configure the monitor
3. Discover the servers (WebLogic and WebSphere)
4. Choose the servers (WebLogic and WebSphere)
5. Process the Server (WebLogic and WebSphere)
6. Discover the counters

7. Choose the counters for the monitoring task
8. Choose the instances of the counter to monitor
9. Review the monitor definition
10. Save and create the monitoring task

Using an Existing Monitoring Task

To select an existing monitoring task:

1. From the Conductor Start Page, click Configure Monitoring in the Tasks area.

   OR

   In the Conductor's Visual Designer, click Tools>Monitor Tasks to open the Manage Monitoring Tasks window.
2. Click File>Open. The Choose an Existing Task dialog box appears.
3. Select a task and click OK. The task displays in Manage Monitoring Tasks window.
4. Select Enable runtime monitoring at the bottom of the window to enable the monitoring task.

Note: You also can enable monitoring in the Session node of the Visual Designer. Use the drop-down arrow in the Monitor task field to select an existing task, then select Enable monitoring. You can open the Manage Monitoring Tasks window to edit or create a task by clicking the browse button next to the Monitor task field.

Adding a Monitoring Machine

Use this procedure to add a monitor to an existing task.

Note: (WebLogic and WebSphere) In a managed server environment, you only can add a monitor to a task from a different administrative server.

- ! For WebLogic, you must use the same WebLogic jar files and WebLogic version as the current monitor.
- ! For WebSphere, you must use the same WebSphere Home, WebSphere client version, and WebSphere server version as the current monitor.

To add a monitor to an existing task under the same administrative server, use Edit an Existing Server Group.

To open the New Monitoring Task wizard:

1. In the Conductor, click Tools>Monitor Tasks to open the Manage Monitoring Tasks window.
2. Click Actions>Add monitor. The Add Monitoring Machine wizard appears. Click Next to start the procedure.

Note: You can open the Manage Monitoring Tasks window to edit or create a task by clicking the browse button next to the Monitor task field.

To add a monitoring machine:

Use the following steps in the Add Monitoring Machine wizard to add a monitoring machine to the task:

Note: (WebLogic and WebSphere) When QALoad detects a managed server environment, you must also select the individual servers to monitor.

1. Enter properties of the monitoring machine
2. Configure the monitor
3. Discover the Servers (WebLogic and WebSphere)
4. Choose the Servers (WebLogic and WebSphere)
5. Process the Server (WebLogic and WebSphere)
6. Discover the counters
7. Choose the counters for the monitoring task
8. Choose the instances of the counter to monitor
9. Review the monitor definition
10. Save and create the monitoring task

Note: See Setting Up Integration with ServerVantage for the procedure used for this monitor type.

Editing a Monitoring Machine

To open the Edit Monitoring Machine wizard:

1. In Conductor, click Tools>Monitor Tasks to open the Manage Monitoring Tasks window.
2. Select a monitor in the Monitors panel, then click Actions>Edit monitor. The Edit Existing Monitor wizard appears.
3. Click Next to start the procedure.

Notes:

! You can open the Manage Monitoring Tasks window to edit or create a task by clicking the browse button next to the Monitor task field in the Session node.

! (WebLogic and WebSphere) In a managed server environment, the Edit Monitor dialog box appears. Do one of the following:

- o Select Edit Server Group and click OK. The Edit Existing Monitor Group wizard appears. Use this option to add, edit, or remove monitors under the same administrative server.
- o Select Edit This Server and click OK. The Edit Existing Monitor wizard appears. Use the procedure for Editing a Single Server in a Managed Server Environment to edit the counters or instances of a single monitor in the managed server group.

To edit a monitoring machine:

Use the following steps in the Edit Existing Monitor wizard to change the properties of a monitoring machine:

1. Enter properties of the monitoring machine
2. Configure Monitor Dialog
3. Discover the Counters
4. Choose Counters
5. Choose Instances
6. Review Monitor Definition
7. Summary

Note: See Setting Up Integration with ServerVantage for the procedure used for this monitor type.

Editing an Existing Server Group

(WebLogic and WebSphere) In a managed server environment, you can add, edit, or remove monitors managed under the same administrative server using this procedure.

To open the Edit Existing Monitor Group wizard:

1. In the Conductor, click Tools>Monitor Tasks. The Manage Monitoring Tasks dialog box appears.
2. Select a monitor, then click Actions>Edit Monitor. The Edit Monitor dialog box appears.
3. Select Edit Server Group, then click OK. The Edit Existing Monitor Group wizard appears.
4. Click Next.

Note: You can open the Manage Monitoring Tasks window to edit or create a task by clicking the browse button next to the Monitor task field.

To add, edit, or remove a monitor:

Use the following steps in the Edit Existing Monitor Group Wizard:

1. Enter Properties of the Monitoring Machine
2. Configure Monitor Dialog
3. Discover Servers
4. Choose Servers
5. Process Server
6. Discover Counters
7. Choose Counters
8. Choose Instances
9. Review Monitor Definition
10. Summary

Editing a Single Server in a Managed Server Environment

(WebLogic and WebSphere) Use this procedure to edit the counters and instances for a single server in a managed server group.

To open the Edit Monitoring Machine wizard:

1. In the Conductor, click Tools>Monitor Tasks. The Manage Monitoring Tasks dialog box appears.
2. Select a monitor, then click Actions>Edit Monitor. The Edit Monitor dialog box appears.
3. Select Edit This Server, then click OK. The Edit Existing Monitor wizard appears.
4. Click Next.

Note: You can open the Manage Monitoring Tasks window to edit or create a task by clicking the browse button next to the Monitor task field.

To edit the server:

Use the following steps in the Edit Existing Monitor wizard:

1. Discover the counters
2. Choose counters
3. Choose instances
4. Review Monitor Definition
5. Summary

Editing Instances

Use the following procedure to edit instances if the counters you are monitoring:

Step 1: Open the Edit Instances dialog box

Step 2: Choose the instances to monitor

Step 3: Review the monitor definition

Step 4: Save the task

Step 1: Open the Edit Instances dialog boxes:

1. In the Conductor, click Tools>Monitor Tasks to open the Manage Monitoring Tasks window.
2. Select the machine, the counter, or the instance to edit.
3. Click Tools>Monitoring>Edit instances. The Edit Instances dialog box displays.

Step 2: Edit the instances of a counter:

1. In the Choose Instances dialog box, double-click the counter group in the left-hand pane to display the counters.
2. Select a counter and click Edit. The Select instance for counter dialog box appears.

Note: When a counter can not be edited, the Edit button is unavailable.

3. Perform the necessary edits. You can do the following:
   - In the Available Instances pane, select an instance and click Add. The instance is added to the Selected Instances pane. Repeat until you select all instances of the counter that you want to apply to the task.
   - In the Selected Instances pane, select an instance and click Remove. The instance is removed from the Selected Instances pane and added to the Available Instances pane.
4. Click Save. The Choose Instances dialog box displays again.
5. Repeat this process for each counter you want to edit.
6. Click Next. The Review Monitor Definition dialog box displays.

Step 3: Review the monitor definition:

1. Review the information for the monitoring machine you defined.
2. Select one of the following:
   - Set up another monitor for this task - returns to the Define Monitor dialog box so you can add another monitor to the monitoring task.
   - Continue without adding any more monitors - continues in this dialog box.

3. (Optional) Click Save as Template to create a template for this monitoring task.
4. (Optional) Select a monitor in the Monitors pane and click Remove Monitor to delete a monitor from the task.
5. (Optional) Type a new value in the Sample Interval field. This is the frequency, in seconds, at which QALoad requests data from ServerVantage during runtime data collection.
6. Click Next. The Summary dialog box displays.

Step 4: Save the task:

1. Review the monitors and counters you have selected for the task. Click Back to return to a dialog box and make changes to the information.
2. In the Monitoring task name field, type a name for the monitoring task.
3. In the Description field, type a description for the task.
4. Select a monitor in the Monitors pane, and click View Monitor Details. The Properties of dialog box displays with detailed information about the monitor configuration and the counters you selected.
5. Click Finish to create the monitoring task.
6. Select Enable runtime monitoring at the bottom of the window to enable the monitoring task.

Note: You also can enable monitoring in the Session node of the Visual Designer. Use the drop-down arrow in the Monitor task field to select an existing task, then select Enable monitoring. You can open the Manage Monitoring Tasks window to edit or create a task by clicking the browse button next to the Monitor task field.

Removing a Monitor or a Counter from a Monitoring Task

Remove a monitor or a counter from a monitoring task, by following this procedure.

To remove a counter from a monitoring task:

1. In the Monitors pane of the Manage Monitoring Tasks window, select the monitor, counter, or counter family to delete.
2. Click Actions>Remove Monitor/Counter.
3. When the verification dialog box displays, click OK.

Note: You cannot remove the last monitor on a machine, the last counter in the family, or the last family of counters in the task.

Monitoring Managed Server Environments

Monitoring Managed Server Environments

WebLogic and WebSphere servers can be configured into managed server environments, where a multi-server group is managed by an administrative server. You can create tasks and templates for managed server environments using the procedures for creating and editing monitoring tasks and selecting the servers from which to extract data.

When you select a WebLogic or WebSphere server to monitor and QALoad detects a managed server environment, it automatically queries the administrative server for the individual servers it manages. All servers that QALoad discovers are listed as available servers that you can select for monitoring. You also can create a template in a managed server environment by selecting an individual server on which to model the template.

Edit the counters and instances for a single server in a managed server group using the procedures for creating and editing monitoring tasks. You also can add, edit, or remove monitors managed under the same administrative server.

Creating Monitoring Tasks for Managed Servers

Use the New Monitoring Task wizard to create monitoring tasks for individual servers or groups of servers in a managed server environment. When you create a monitoring task for a managed server environment, you select an administrative server to monitor. QALoad queries the administrative server for the individual servers it manages. All servers discovered are listed as available for monitoring.

> Note: When the information returned by the administrative server indicates that it does not manage any other servers, the counter discovery process begins for the individual administrative server.

Once you select the servers to monitor, QALoad begins the counter discovery process for each server in turn. Select the counters and instances of counters to monitor on the first server. QALoad includes these in the task, and then begins the counter discovery process for the next server you selected.

Creating Monitoring Templates for Managed Servers

You can select counters and instances and save them to a template that you can use for other monitoring tasks.

When you create a template in a managed server environment, you select a server on which to model the template and adding the counters and instances of counters to the template.

QALoad queries the administrative server for all the servers it manages. From this list, select the server to use as the model when creating the template. QALoad's New Monitoring Template wizard guides you through the process of adding the counters and instances of counters to the template.

> Note: You can select only one server as a model for the template. If the server you select is unavailable, you are returned to the managed server selection dialog box to choose another server.

Editing Monitors in a Managed Server Environment

In a managed server environment, you can edit a single server or modify an existing group of servers in a task.

Edit an individual server using the Edit Existing Monitor wizard. You can add or remove counters and instances to monitor on the server. When you edit a server group, you use the Edit an Existing Server Group wizard. You can add monitors, edit the properties of a monitor, or remove monitors managed under the same administrative server. You also can change the counters and instances to monitor on an individual machine.

ServerVantage

Overview of Server Monitoring with ServerVantage

If you are currently a licensed user of Compuware's ServerVantage, you can integrate data from your existing ServerVantage deployment directly into a QALoad timing file.

For this method to be successful, the following conditions must be met:

- ServerVantage must be installed and configured correctly on your system.
- ServerVantage must be scheduled to monitor the specified performance counters at a time that coincides with a running QALoad test.

! You must configure the port to use for the SQL database. The port must be open on the ServerVantage database server so that QALoad can retrieve the counter data at the conclusion of the test. The default SQL port is1433.

! QALoad must be able to access the ServerVantage database server on port 139 or 445 via tcp to obtain time stamps at the beginning and end of the test.

! QALoad must be able to access the ServerVantage agent using an ICMP ping during the monitor setup. If security restrictions prevent pinging the agent, an entry can be added to the host's file on the Conductor machine mapping the domain name of the agent to the IP address of a machine that can be pinged, such as the Conductor.

About ServerVantage

ServerVantage (formerly EcoTOOLS) monitors the availability and performance of applications, databases and servers, allowing users to centrally manage events across all application components— Web servers, firewalls, application servers, file systems, databases, middleware, and operating systems. ServerVantage simultaneously monitors these components, analyzes both historical and real-time events, and correlates monitored information for problem detection.

Integration with ServerVantage is configured from the QALoad Conductor. Performance counters collected during a load test are included in the test's timing file and can be sorted and displayed in QALoad Analyze in much the same way as QALoad timing data. For more information about installing or configuring ServerVantage, refer to its product documentation.

Setting Up Integration with ServerVantage

Use the following steps to set up integration with ServerVantage:

Step 1: Open the New Monitoring Task Wizard

Step 2: Define and Configure the Monitor

Step 3: Review the Monitor Definition

Step 4: Review the Summary and Create the Task

Step 1: To open the New Monitoring Task wizard:

1. Click Tools>Monitor Tasks.
2. Click the Set up monitoring link, then select Set up a new monitoring task, then click OK to open the New Monitoring Task wizard. Click Next.

Step 2: To define and configure the monitor:

1. In the Define Monitor dialog box, click the arrow in the Monitor Type box and select ServerVantage.
2. In the Control Server Database Host field, click the down arrow and select the hostname of the machine where the ServerVantage server is located.
3. Click Next. The Configure Monitor dialog box displays.
4. In the Username field, type a valid user name to access the ServerVantage server, if necessary.
5. In the Password field, type the password that corresponds to the user name above, if necessary.
6. Select the Override Default Database check box to provide the ServerVantage database name. When this option is not selected, QALoad uses the default ServerVantage database name. If you

provided a different name during the installation of ServerVantage, select this option and type the name in the Database Name field.

7. In the Name field in the Vantage Agent area, type the hostname of a machine(s) where a ServerVantage Agent is installed, and click the Add button to add it to your load test.
8. Click Next to proceed to the next step, Review Monitor Definition.

Step 3: To review the monitor definition:

1. Review the information for the monitoring machine you defined.
2. Select one of the following:
   - Set up another monitor for this task - returns to the Define Monitor dialog box so you can add another monitor to the monitoring task.
   - Continue without adding any more monitors - continues in this dialog box.
3. (Optional) In the Monitors pane, select the monitor, then click Save as Template to create a template for this monitoring task.
4. (Optional) In the Monitors pane, select the monitor type, then click Remove Monitor to delete a monitor from the task.
5. (Optional) Type a new value in the Sample Interval field. This is the frequency, in seconds, at which QALoad requests data during runtime data collection.
6. Click Next to proceed to the next step, where you review the summary and create the task.

Step 4: To review the summary and create the task:

1. Review the monitors and counters you have selected for the task in the Summary dialog box. Click Back to return to a dialog box and make changes to the information.
2. In the Monitoring task name field, type a name for the monitoring task. The task is saved so you can reuse this configuration of counters and instances.
3. In the Description field, type a description for the task.
4. Select a monitor in the Monitors pane, and click View Monitor Details. The Properties of dialog box displays with detailed information about the monitor.
5. Click Finish to create the monitoring task. The Manage Monitoring Tasks window displays.

Displaying ServerVantage Agent Data

If you set options to integrate ServerVantage resource utilization data before running a test, that data is included in the resulting timing file. It can be sorted and displayed in QALoad Analyze in much the same way as QALoad timing data. ServerVantage data provides a summary of all the Agents that ServerVantage monitored during the load test and details aggregate statistics for Agent data points including minimum, maximum, and mean data values.

When you open a timing file containing ServerVantage Agent data, QALoad Analyze displays test data with QALoad timing data two ways:

- ServerVantage Agent workstations are listed in the Server Monitoring group in the Workspace tree-view, under the Resource Trends (ServerVantage) branch. From the Workspace, select Agent workstations to create detail or graphical views of the Agent data points. Specifically, you can:
  - Display Agent data point details.

" Graph Agent data point details.

! Detailed data point information is displayed in the Data window. The ServerVantage detail view includes data such as the name of the machine where you ran the ServerVantage Agent; the Agent name; and the minimum, maximum, and mean data values for the Agent.

Note: ServerVantage resource utilization data is available only if you set the ServerVantage integration options on the QALoad Conductor's Test Information window before executing a load test.

ApplicationVantage

Overview of ApplicationVantage

QALoad integrates with ApplicationVantage to help you analyze network performance during a load test. ApplicationVantage provides granular thread details that allow network managers to identify poorly performing applications. QALoad also provides test data that you can open in ApplicationVantage.

Before QALoad can collect network data during a load test, the following must be true:

! The ApplicationVantage Agent is installed on the same machine as the QALoad Conductor. You can install either the ApplicationVantage Agent or the ApplicationVantage Remote Agent.

! You have specified on which NIC to capture in the Manage Players/Groups dialog box in Conductor. How?

At test time when a transaction is started, the Player configured to capture ApplicationVantage data starts an ApplicationVantage trace. The trace stops when the transaction completes. When a Player is running a script that is set to run in ApplicationVantage mode, every transaction generates a new trace file. At the end of the test, these files are packaged into the test's timing file.

Hint: For information about ApplicationVantage, refer to the documentation you received with your purchase of this tool.

Configuring a test to use ApplicationVantage

Integration with ApplicationVantage enables you to study network problems in detail. You can set up one or more ApplicationVantage (AV) Player machines for the load test. These AV Player machines run a QALoad script on a periodic basis while the AV Agent captures the network traffic that the script produces. The resulting AV trace files (*.opx) are sent back to the Conductor with the regular QALoad timing file for analysis after the test is complete.

To enable ApplicationVantage, you must be running ApplicationVantage 10.0 or greater. You must enable ApplicationVantage in the Properties window, and set the Network Interface Card (NIC) Name used by the machine on which the data is captured.

Enabling ApplicationVantage

You can enable or disable the ApplicationVantage for each load test on a script. To enable ApplicationVantage, you must select the option, and then set the Network Interface Card (NIC) Name.

To enable ApplicationVantage:

1. Click the script icon for the appropriate script in the Visual Designer window. The Script Properties panel appears on the right-hand side of the window.
2. In the Script Properties panel, click the ApplicationVantage field, then select True.

3. Click Tools>Manage Players.
4. If necessary, set the NIC Name.

Setting up Network Interface Card Name

To use the ApplicationVantage Agent to collect data for ApplicationVantage, it is necessary to specify which Network Interface Card (NIC) to capture on. This is the network information for the workstation where your ApplicationVantage Remote Agent is installed.

To set up NIC Name:

1. On the Conductor's toolbar, select Tools>Manage Player. The Manage Players/Groups dialog box displays with names of available Player machines listed in the Players area.
2. Click the Player machine that will be running the virtual user to be captured. The information for that Player machine displays in the Player Information area.
3. If necessary, click the ⊞ button next to Application Vantage Settings to expand the information.
4. From the drop-down list in the NIC Name field, select the NIC that is used by the machine.
6. Click Save, then click OK.

ClientVantage

Overview of ClientVantage

ClientVantage manages end-user application performance and availability. Problems can be diagnosed by powerful fault detection and analysis capabilities as well as resource monitoring.

ClientVantage must be installed on the same Windows workstation as the QALoad Conductor and the QALoad Player.

Vantage Analyzer

Vantage Analyzer Integration

Vantage Analyzer is designed for easy resolution of complex application performance issues. It enables you to easily drill into specific problem transactions to determine the cause of bottlenecks in your production applications. It also enables you to find Java code and SQL statements that are consuming excessive resources. Troublesome memory leaks that are observed in your actual production servers can be quickly resolved.

If you are currently a licensed user of Compuware's Vantage Analyzer, you can integrate data from your existing Vantage Analyzer deployment directly into a QALoad timing file.

For this method to be successful, the following conditions must be met:

! Vantage Analyzer 10.1 SP1 must be installed and configured correctly on the same machine as QALoad Analyze. For more information about installing or configuring Vantage Analyzer, refer to its product documentation.

! Time has to be synchronized between the QALoad Conductor machine and the Vantage Analyzer Nucleus Server machine to make testing data more meaningful. The difference of the time between the two machines is saved in a timing file.

Setting Up Integration with Vantage Analyzer

To set up integration with Vantage Analyzer:

1. Open or create a session in the Conductor. From the Tools menu, choose Monitor Tasks. The Manage Monitoring Tasks dialog box displays.
2. Click the Set up monitoring link and select Set up a new monitoring task. Click OK. If the Welcome to the New Monitoring Task Wizard appears, click Next.
3. In the Define Monitor dialog box, click the arrow in the Monitor type box and select Vantage Analyzer.
4. In the Nucleus Server Name or IP address field, type or select the machine host name or IP address of the machine where the Vantage Analyzer Nucleus server runs.
5. Click Next. The Configure Monitor Dialog box displays.
6. In the Username field, type the login user ID for the Vantage Analyzer Nucleus server machine (not the Nucleus server itself).
7. In the Password field, type the login password for the Vantage Analyzer Nucleus server machine (not the Nucleus server itself).
8. Click Next. The Review Monitor Definition dialog box displays.

Index