QALoad 5.5 Help

Getting Started, Running a Load Test, Analyzing Test Results, and 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

This document and the product referenced in it are subject to the following legends:

Table Of Contents

QALoad Online Help

Welcome to QALoad

With QALoad, you can simulate the load generated by thousands of users without the expense of actual end users or their equipment. QALoad enables you to quickly develop test scripts, control the conditions for the test, create the virtual users that simulate the load, initiate and monitor the test, and report the results.

The following components of QALoad control different aspects of the load testing process:

Script Development Workbench
Develop your scripts. Capture sessions and convert the transactions into C, C++, or Java-based script.

Conductor
Control session activity. Initiate and monitor the test.

Player
Simulate user activity based on commands given from the Conductor.

Analyze
View summary report data and create other statistical reports from the test.

Before you begin using QALoad, review the Getting started section of the online help to familiarize yourself with load testing concepts.

Getting started

The load testing process

You begin the testing process by determining the types of application transactions you want to emulate. You then develop these transactions into QALoad scripts by creating the same types of requests that your applications invoke on the server. Each transaction becomes its own script. The QALoad Script Development Workbench lets you easily create full-function scripts.

When you plan your test, you need to decide which transactions to run, the number of simulated clients that will run each transaction script, and the frequency at which each script will run. When you run the test on a workstation with the QALoad Player component, you can specify transaction rates as well as fixed and random delays to better emulate real-system activity. QALoad considers these factors a test scenario and stores them in a session ID file.

While a test is running, the test operator can dynamically view overall run times as well as individual transaction performance. QALoad's Conductor component collects this data for analysis at the conclusion of the test.

After executing the test, summary reports show the response times that the emulated clients experienced during the test. Individual and global checkpoints let you view and identify specific areas of system performance. You can export all test output data to spreadsheet and statistical packages for further analysis or use QALoad's Analyze component to create presentation-quality reports and graphs.

As shown in the following image, a typical load test setup consists of a QALoad Conductor, one or more QALoad Players, and the system under test.

QALoad Conductor

QALoad's Conductor controls all testing activity such as setting up the session description files, initiating and monitoring the test, and reporting and analyzing test results.

QALoad Player

A QALoad Player creates virtual users that simulate multiple clients sending middleware calls to a server under test. In a typical test setup, one or more QALoad Player workstations run under any Windows 32-bit platform (Windows XP or 2000) or UNIX. For large tests (thousands of simulated clients), you can connect multiple Players to QALoad's Conductor. The Conductor and Players communicate using TCP/IP.

The hardware and software capabilities of the Player machine are the only factors that limit the capacity of an individual QALoad Player. The maximum number of virtual users per Player machine is dependent on the system under test, the characteristics of the script, and the test scenario. You can specify how many threaded- and process-based virtual users to assign on a machine on the Machines tab of the Options dialog box in the Conductor. The Conductor calculates how many virtual users will be active per 64 MB of RAM, based on the values you provide in these fields.

System under test

The servers you test are typically production systems or a duplicate of a production system that is set up at a test facility. If you perform any kind of system selection or performance stress test, the system under test must use the same hardware and software (including current versions) as the production environment. Compuware has found that even subtle changes have profound effects on performance.

Developing scripts

You use the QALoad Script Development Workbench to develop test scripts. It contains facilities for capturing sessions, converting captured sessions into scripts, and modifying and compiling scripts. Once you compile your script, you can use QALoad's Conductor and Player components to test your system.

Record Facility: QALoad's Record facility, which you can access through the QALoad Script Development Workbench, records the transactions that your terminal, browser, or client makes. It stores these transactions in a capture file.

Convert Facility: QALoad's Convert facility, which you can access through the QALoad Script Development Workbench, converts capture files into scripts. It generates a one-to-one correspondence of transactions from the original session to your QALoad script.

Visual Navigator: Visual Navigator for WWW is QALoad's easy-to-use visual interface to QALoad's powerful script development tools. Visual Navigator for WWW renders your recorded C-based transaction in a tri-paned, browser-like environment similar to popular visually-oriented development tools, with icons representing all the elements of your script.

Using QARun scripts for load testing

QALoad provides you with the functionality to perform load tests using your QARun scripts. By inserting your QARun script into a QALoad script template, you can time your GUI-driven business transactions and include those timings in QALoad post-test reports.

Validating scripts

Before you conduct an actual load test, you should individually validate the script(s) you plan to use in the load test by running it in a simple test. If the script runs to the end without any errors and runs multiple times without errors, it is valid to use in a load test.

If the script aborts on an error, debug the script and run it through a simple test again. You can validate scripts from the QALoad Script Development Workbench, the QALoad Player, or the QALoad Conductor:

Validate a script from the Script Development Workbench

Validate a script from the Player

Validate a script from the Conductor

Setting up the Conductor

To prepare for running a load test, you must 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. You use the Conductor to create and save session ID files in the `\QAload\Session` directory.

Server and performance monitoring

QALoad integrates several mechanisms for merging load test response time data with server utilization data and performance metrics. Most methods produce data that is included in your load test timing results and processed in QALoad Analyze. The only exception is ApplicationVantage. Data captured from ApplicationVantage can be opened in ApplicationVantage, but not in QALoad.

If you plan to collect server and performance monitoring data, you must set the appropriate options in the Conductor before running the load test. The following methods of server and performance monitoring are available:

- Remote Monitoring — allows you to monitor server utilization statistics from a remote machine without installing any software on the remote machine.
- Server Analysis Agents — must be installed on each applicable 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 *QA*Load .
- ApplicationVantage — collects test data that you can open in Application Vantage.

Starting the Conductor

The following procedure describes how to start the Conductor.

To start the QALoad Conductor:

1. **From Windows:**
   - Click Start>Programs>Compuware>QALoad>Conductor.

   From a Command Prompt:
   - Type `mpwin32 <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`.

2. When the **Session Options** dialog box opens, select the appropriate option to create a new session ID file or to open an existing one. Click **OK**.

3. If you chose to open an existing session ID file, the **Open** dialog box appears. Navigate to the session ID file you wish to open, select it, and click **Open**. The selected session ID file opens in the Conductor Test Information Window.

If you chose to set up a new session ID file, the Conductor opens displaying the Test Information Window, and the configuration and setup toolbar. For information about the Conductor's configuration and setup toolbar buttons, see Configuration and setup toolbar buttons.

Running a Load Test

Running a load test

After validating a script using one of the methods described in Validating scripts, it is safe to run a load test with that script. See the following topics for more information:

- Preparing for a Load Test
- Starting a Load Test
- Monitoring a Load Test
- Stopping a Load Test

Preparing for a load test

Before you run a load test, you must complete the following tasks:

- **Prepare a datapool file:** If you created a datapool file using the *QA*Load Script Development Workbench, *QA*Load stores the file where the Conductor can automatically access it. However, if you created a datapool file using a text editor (for example, Notepad), you must place the file in your appropriate `\Middlewares\<middleware_name>\scripts` directory (for example, `\QAload\Middlewares\Oracle\Scripts`) so the Conductor can access the file.

  For information about datapool files, see Simulating user-entered data.

- **Set Up SSL Client Authentication for Virtual Users (SSL scripts only):** If you are running a load test with a WWW script containing SSL requests, you should export a Client Certificate from your browser into *QA*Load or create a *QA*Load Client Certificate for each virtual user that runs the script. This setup facilitates a one-to-one ratio of Client Certificates to virtual users, which more realistically simulates your testing environment.

  To export Client Certificates from your browser and convert them for use in *QA*Load or to create *QA*Load Client Certificates, see Importing a client certificate from a Web browser.

  Once you export or create the necessary Client Certificates, you can insert them into your script using datapools.

Starting a load test

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 about what to expect from the QALoad Conductor while a test is running — including descriptions of the Runtime Toolbar buttons — see Monitoring a load test.

To start a load test, click the Run button on the configuration and setup toolbar or choose Start Test from the Conductor's Run menu.

> 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 impact the test results.

Monitoring a load test

When a test is started, the QALoad Conductor's interface changes to an interactive test control station, referred to as the Runtime Window. The Runtime Window displays information about the scripts, machines, and virtual users that are executing the load test. From the Runtime Window, you can observe the progress of individual scripts and Player machines, create and view real-time graphs, and start or suspend scripts and Players from a running test to better simulate the unpredictability of real users.

In addition to the test data shown by default on the Runtime Window, you can access detailed test information using the QALoad Conductor’s Runtime toolbar buttons. You can:

- View statistics for a single virtual user
- View the activities of a virtual user in a browser-like window (WWW only)
- Step to the next request (WWW only)
- View the current datapool record
- Display the script running on a single virtual user
- Display messages sent from a Player workstation to the *QA*Load Conductor
- Display statistics about Conductor/Player communication
- Show/hide the Runtime Tree or Runtime Control Panel
- Synchronize all virtual users
- Exit, abort, or quit the test

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. This means the .c files referenced in the selected session ID files must be present in the scripts directory.

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.

To stop a load test, click the Exit button.

Adding post-test comments

If you selected the Display Post Test Comments option on the General tab of the Options dialog box when you configured the Conductor, the Post Test Comments window opens when you click the Quit button. Type any comments, which are saved to the test’s Summary Report, which can be viewed in QALoad Analyze.

Analyzing Test Results

Analyzing test results

After you set up a load test and run it, you can analyze the results from the test using QALoad Analyze.

An important part of the load testing process is viewing and studying the results of a test. You can view load test results not only on a machine where QALoad is installed, but also on any machine with a Web browser.

When you run a test using a particular session ID file (set up in the Conductor), each Player compiles a local timing file comprised of a series of timing records for each checkpoint of each script run on that Player. Each timing record in the file consists of a response time/elapsed time pair of values specifying the amount of time it took a certain checkpoint to finish (response time) at a specific time in the test (elapsed time).

At the end of a test, Player timing files are sent to the Conductor and are merged into a single timing file, called the Primary timing file, for analysis. If you set up integration with Compuware's ServerVantage product, the Conductor collects timing data from the ServerVantage central console and also merges that data into the timing file.

When you open a timing file, QALoad generates a working folder that contains all supporting files, reports, and images generated from that timing file. The folder is located in the `\Program Files\Compuware\QALoad\TimingFiles\xxx.xml.source` directory, where <xxx> is the name of the timing file.

Custom reports

QALoad Analyze provides the ability to create custom reports using XML (Extensible Markup Language), XSL (Extensible Style Language), and HTM (Hypertext Markup) files. QALoad Analyze provides a set of files in .htm, .xml, and .xsl formats in addition to the .tim file. QALoad Analyze automatically generates a XML (*.xml), XSL (*.xsl), and HTM (*.htm) file when you open a timing file.

Pre-defined reports

QALoad Analyze provides pre-defined reports so you can receive immediate load test results without having to manipulate any data. All the files necessary for those reports are located in the directory `\Program Files\Compuware\QALoad\Timing Files\Reports`.

Note: The pre-defined reports that are available depend on the data collected in the timing file, which is determined by the QALoad Conductor option you select at the time of running the load test.

Graphing data

Starting with the Workspace, you can use QALoad Analyze's charting features to graph timing data in a number of formats and styles.

Managing large amounts of data

With a large number of virtual users, it's 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 QALoad Analyze down 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 has to 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, right-click on the graph, or so on, 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.

QALoad Analyze graph types

The following basic graph types are available from QALoad Analyze. After generating one of the following graph types, you may further customize a graph in a number of ways.

Line Graph

A line graph plots response times versus elapsed times for the selected checkpoints. It provides a good representation of how much fluctuation there is in response times over the course of a test.

Bar Graph

A bar graph shows the median, mean, or percentile response times for the selected checkpoints.

Transaction Throughput Graph

This type of graph shows the cumulative number of transactions that occurred within the user-specified time range over the duration of the test.

Response Time Distribution Graph

This type of graph shows the percentage of checkpoint timings that fall within a particular response time range. A response time distribution graph shows if response times tend to fall within a range or are widely dispersed. A response time distribution graph only shows results for a single checkpoint, although it can compare results from multiple timing files.

Cumulative Response Time Distribution Graph

This type of graph shows the percentage of transactions for a single checkpoint that have a response time equal to or less than a specified value.

Customizing graphs

You can also change the style and appearance of a graph using options available from the Graph toolbar. Display the Graph toolbar by right-clicking in an open area of a graph and choosing the Toolbar option from the shortcut menu. The Graph toolbar contains buttons for standard Windows operations as well as for customizing the appearance of your graphs.

The Graph toolbar includes the following features for customization:

- Graph type (gallery type)
- Color
- Grid orientation (horizontal and vertical)
- Legend box
- Data display
- Dimension (3D or 2D)

- Rotation
- Z-Cluster
- Color/pattern

Integrating 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.

Displaying ServerVantage Agent data

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.

Conductor

About the Conductor

Overview of the QALoad Conductor

You use the QALoad Conductor to configure, run, and monitor a load test that utilizes the scripts you created in the Script Development Workbench. The Conductor controls the QALoad Players and manages tests while they're running.

Before running a test, you must set up a test by recording descriptive information about the test, setting general test options, configuring Player workstations, assigning compiled test scripts to Players, and setting up monitoring options. Then, save the test setup in a file called a session ID. Once you've configured and saved a test session ID, you can reuse it without re-entering any of your test information.

While a test is running, the Conductor interface changes to a tri-pane view called the Runtime window that 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. Your custom graph layouts can be saved in the session ID file and reused in future tests.

Taking a look at the Conductor

The Conductor's interface is dynamic — it changes depending on where you are in the testing process: setting up a test, or running a test. Both interfaces are described below.

Test setup

The Conductor's Main Window is divided into tabs on which you enter information about your test and set up the machines and scripts for the test. For more information about the test setup interface, see Test setup interface.

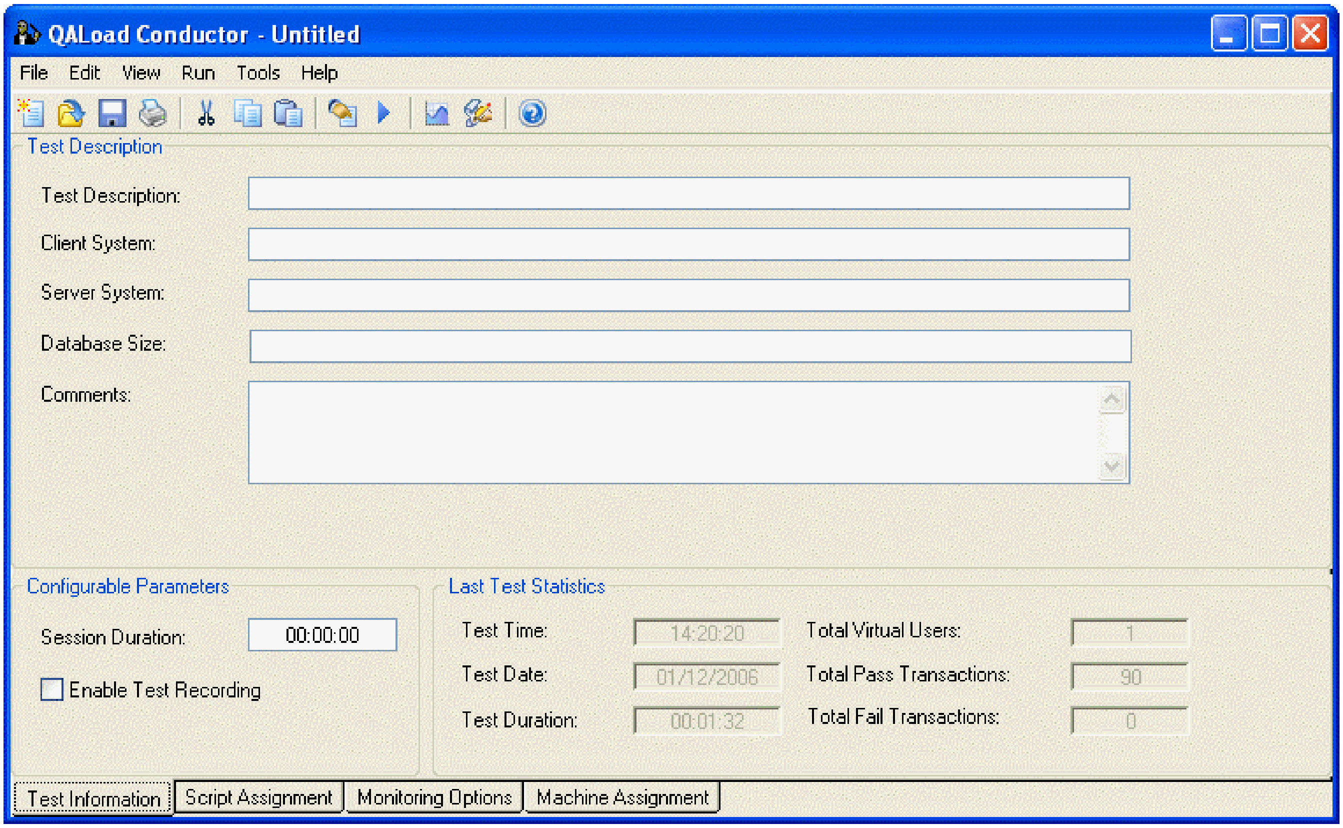
Running a test

While a test is running, the Main Window changes into the Runtime Window, which displays vital information about your running test and provides several controls to alter your test on-the-fly. For more information about the Runtime Window and how to use it while running a test, see Running a test.

Test Setup

Test setup interface

The Conductor's Main Window is divided into tabs on which you enter information about your test and set up the machines and scripts for the test. In addition, the Conductor'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. Use the tabs on the Main Window to set up your test. For detailed information about the fields on these tabs, click one of the links that follow or press F1 from any Conductor tab.



Test Description

Test Description:

Client System:

Server System:

Database Size:

Comments:

Configurable Parameters

Session Duration: 00:00:00

□ Enable Test Recording

Last Test Statistics

Test Time: 14:20:20

Test Date: 01/12/2006

Test Duration: 00:01:32

Total Virtual Users: 1

Total Pass Transactions: 90

Total Fail Transactions: 0

Test Information

Use this tab to enter descriptive information about the test, view statistics from the previous test, and set a maximum amount of time for the current test to run. All descriptive information about the test will be incorporated into your test's timing file, and can be viewed in Analyze with your test statistics after the test has finished.

Test Description: Enter a description of the test, for example, its purpose. This field is written to your test's timing file to describe the test. This field is optional.

Client System: Enter a description of the client workstations. This field is optional.

Server System: Enter a description of the server(s) under test. This field is optional.

Database Size: Enter a description of the database used in this test. This field is optional.

Comments: Enter any general comments about the test in this field. Note that this field is overwritten at the end of a test if the Post Test Remarks check box on the Options dialog box, General tab is enabled. In that case, notes you type in the Post Test Remarks dialog will be saved to the test's Summary Report.

Session Duration: Enter a value to specify the maximum amount of time for the test to run. Entering zero (the default) will execute the test indefinitely. You can determine whether to stop the test immediately when the duration expires or to allow the virtual users to complete the running transactions by using the option Gracefully exit users when Session Duration expires on the Options dialog box, General tab.

Enable Test Recording: Select this check box to record load tests for later review.

Test Time: This field displays the time the previous test started.

Test Date: This field displays the date the previous test started.

Test Duration: This field displays the length of time it took for the previous test to run.

Total Virtual Users: This field displays the number of virtual users assigned to the previous test.

Total Pass Transactions: This field displays the number of transactions that ran successfully in the previous test.

Total Fail Transactions: This field displays the number of transactions that failed in the previous test.

Script Assignment

Use this tab to set up any scripts that have previously been recorded and compiled. Any scripts you add here is 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.

Script: Select a script from the drop-down list, or click the New button to add a script. Click the browse [...] button to open the Select Script dialog box.

New button: Click this button to browse for a script to add to the test.

Delete button: Click this button to remove the selected script from the load test.

Delete All: Click this button to delete all scripts from this test.

Type: Lists the middleware type of the script. Click the browse (...) button to set custom middleware options.

Transactions: Type the number of transactions that each virtual user running the designated script should run. Once a workstation reaches the maximum number of transactions, the script continues execution with the line following the End_Transaction command rather than returning to the top of the transaction loop. If you enter 0, the Conductor executes the script indefinitely.

Debug Options: Click the browse button to access the Debug Options dialog box, where you can configure debug settings for the script.

Error Handling: Choose how to respond when a non-critical 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:

- Continue Transaction — If an error occurs while a transaction is being executed, the Player should continue executing the transaction as if the error had not occurred. Select this option when errors are not critical to the performance of the load test and can be safely ignored.
- 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.
- 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 increases for each transaction that is restarted.

QALoad uses two commands — DO_SetTransactionStart() and DO_SetTransactionCleanup() — to control error handling. These commands are inserted into your script by the conversion process.

Sleep Factor %: 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.

Valid values for Sleep Factor % are 0-999,999% and also Random, which causes the Player to sleep for a randomly selected duration between 0 and the number of seconds specified in the DO_SLEEP() statements.

When a load test is executed with a Sleep Factor of 100% the script executes 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.

Service Level Threshold: Enter a response time to use as the threshold for comparing other response times. When you run a test, a line representing the Service Level Threshold will appear in the runtime window. As the test progresses, you can compare incoming response time data to the Service Level Threshold. This is a preliminary way of analyzing test results without waiting to open the timing file at the end of the test, and can be used as an indicator to determine if/when Dial-Up users should be added to the test.

Pacing: Enter a value in this field for 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.

The default pacing value is one second. This allows the Conductor to throttle runaway virtual users.

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.

Timing Options: Click the browse button to open the Timing Options dialog box, where you can choose to include QALoad's automatic timings in your test results or determine how much timing data to collect.

External Data: Click the browse button to open the External Data dialog box where you can select any central datapool files, local datapool files, or additional external support files necessary for your test. For additional information, see Using the certDB File for OFS Replay.

Monitoring Options

Use the Monitoring Options tab to specify options for your monitoring task. You also can integrate ServerVantage into your load test. QALoad assumes that the appropriate ServerVantage software is installed, configured, and running before you start a load test.

Enable Performance Monitoring at Runtime: Enables the monitoring of the counters specified in a monitoring task you create.

When you select a monitoring task, the following displays:

Monitors: Displays the list of groups of monitors and associated machines selected for this monitoring task. Use the selection button to choose to display groups by machine name or by monitor type.

Monitors detail: Displays details of the monitoring task you selected including Machine name, Monitor Type, and the date it was last updated.

Use the icon to choose how to view the monitors. You can choose:

- Group by machine name
- Group by monitor type

When you select a monitoring template, the following displays:

Templates: Displays the name of the template you selected.

Counters: Displays the groups and counters defined in the template.

View Properties: Displays the properties for the monitoring task.

Set up monitoring: Displays the Set Up Monitoring dialog box, where you can choose to set up a new monitoring task or use an existing task stored as a template.

Machine Assignment

Use the Machine Assignment tab to assign scripts to specific Player workstations. You can use the Edit menu's Copy and Paste commands to copy and paste machine entries (rows) as needed.

Script: This field displays the script name. To add a script, click New.

Middleware: Displays the middleware type the script was created for. Use the browse [...] button to open the Expert User Options dialog box.

Starting VUs: Type number of virtual users to begin the test.

VU Increment: Type the number of virtual users to be added, at intervals, after the test begins.

Time Interval: Displays the time interval at which incremental virtual users will be added to a test. Change the time interval by typing a new value.

Ending VUs: Displays the number of virtual users assigned to run until the end of the test.

Machine: From the drop-down list, select Player machines to assign each script to run on.
For ApplicationVantage integrations, the Player and Conductor machines must be on the same LAN.

Mode: Select the test mode for each Player machine: thread-based, process-based.

New button: Click to access the Select Script dialog box where you can select a script to add to the test.

Delete: Click to delete the selected script.

Delete All: Click to delete all scripts.

Manage Player Machines: Click this button to display the Manage Player Machines and Groups dialog box.

Auto Configure: Click this button to have QALoad automatically assign scripts to virtual users.

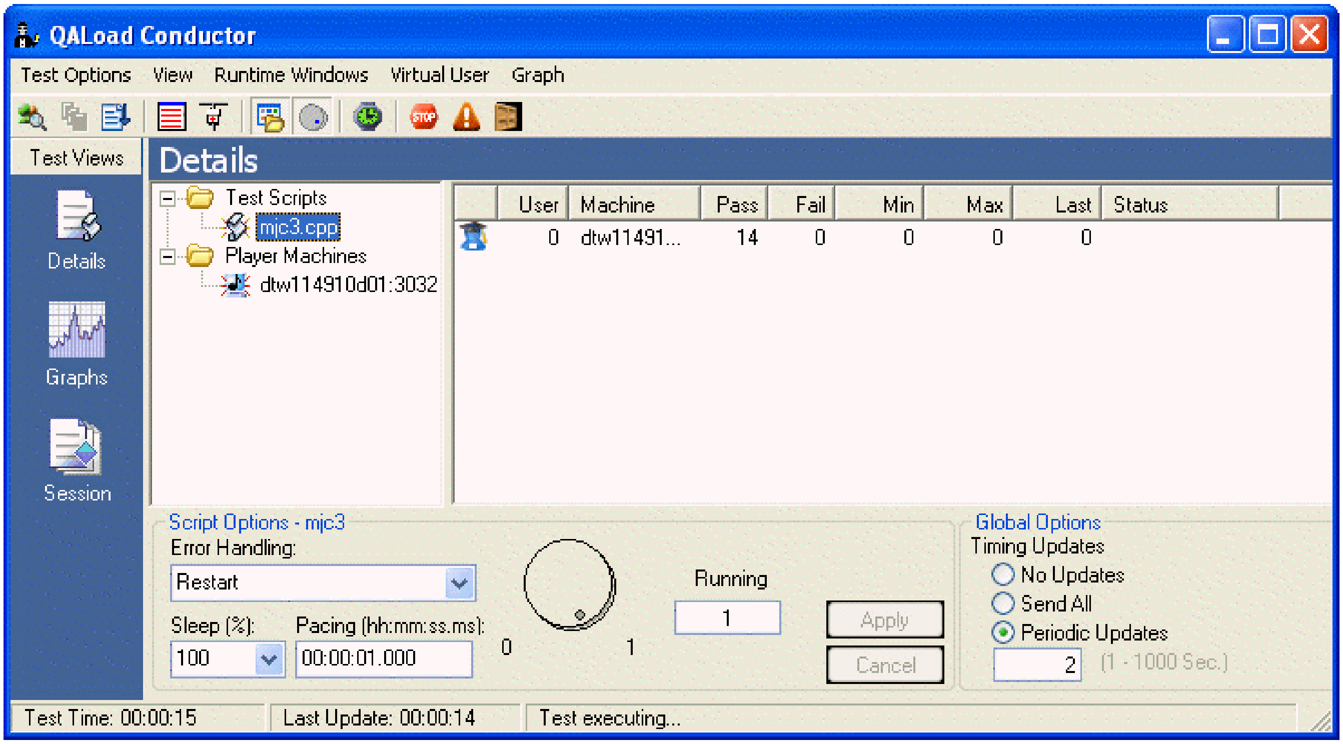
Run: Click to start a test run. This button is only enabled if your test is completely set up.

Running a Test

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 test data is divided into three views – Details, Graphs, and Session – that are accessed from the Test Views workspace on the left side of the interface.

From 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. This window has three unique areas. Click on a pane in the following graphic for a brief description of that pane. Or, click on the links below the graphic for detailed information.



User	Machine	Pass	Fail	Min	Max	Last	Status
0	dtw11491...	14	0	0	0	0	

Error Handling: Restart

Sleep (%): 100

Pacing (hh:mm:ss.ms): 00:00:01.000

Running: 1

Timing Updates:
○ No Updates
○ Send All
● Periodic Updates
2 (1 - 1000 Sec.)

Test Time: 00:00:15 | Last Update: 00:00:14 | Test executing...

Details view

The Details view of the Data window displays all your test data in real-time in a series of interactive tabs. By clicking on icons representing scripts, virtual users, and workstations, you can view different types of data. By default, each test displays test details in the right pane. You can also choose to view the script a single virtual user is running, the Web page a WWW script is utilizing, or the RIP file generated by a failed virtual user.

Test details

Test details display automatically, and describe the object you select in the tree view. You can view details for all test scripts, individual test scripts, all player machines, and individual player machines.

See Test details for more information about the Tree items that can be displayed in the data window.

Runtime tabs

The following runtime tabs can be displayed for a running script. These tabs can be enabled from the Virtual User menu.

Debug
Monitor (Web User)
RIP File (WWW)

Runtime Control Panel

The Runtime Control Panel is a dockable control station that enables you to change virtual user options and data transfer options while the load test is running. For more information, see Runtime Control Panel.

Graphs view

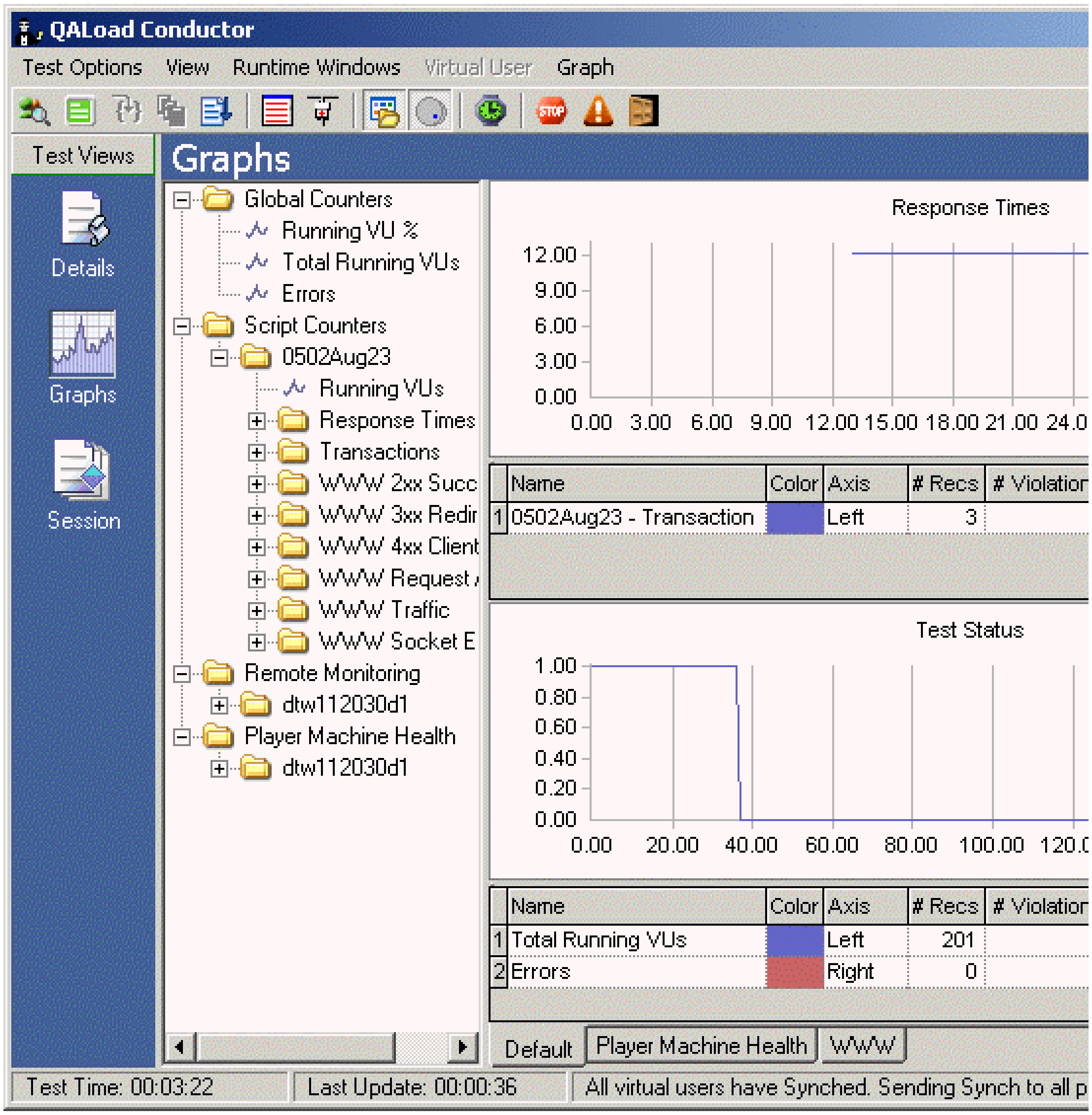
The Graphs view displays graphs of data collected during the test. By default, the Graphs view displays graphs for response times, test status, and player machine health.

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	

	Name	Color	Axis	# Recs	# Violation
1	Total Running VUs	🟦	Left	201	
2	Errors	🟥	Right	0	

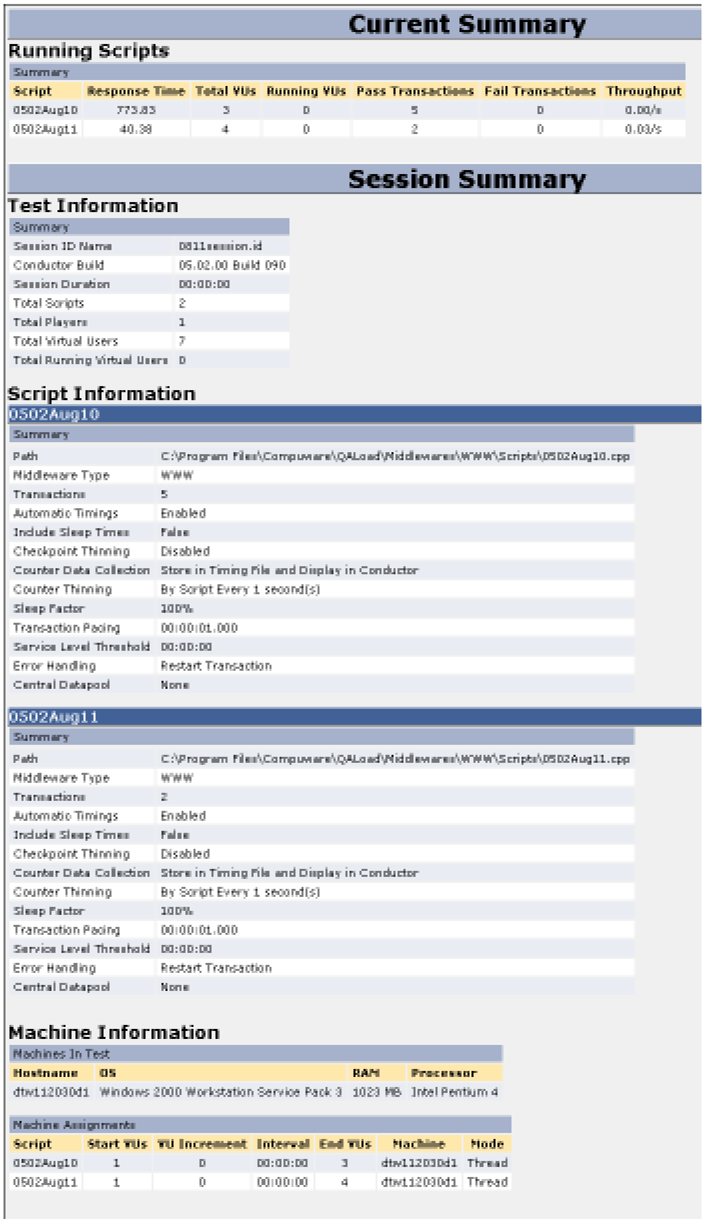
Test Time: 00:03:22 | Last Update: 00:00:36 | All virtual users have Synched. Sending Synch to all p

Session view

The Session view 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

QALoad Conductor menus and toolbar buttons

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

Test setup

The Conductor Configuration and Setup Menus allow you to configure the Conductor and your specific test. Click a menu or toolbar name below for details:

File
Edit
View
Run

Tools
Help
Configuration and Setup toolbar buttons

Running a test

The Conductor's Runtime menus and toolbar allow you to control your running test and the data that is displayed at test time. Click a menu name below for details:

Test Options
View
Runtime Windows
Virtual User
Graph
Runtime toolbar buttons

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 Analyze after the test.

When the Top Process Monitoring counter is enabled for a Server Analysis Agent machine, information is collected periodically during the load test about which processes are using the most CPU. Counter data is written to your test's timing file, which you can open in Analyze after your 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.

To collect Top Processes data:

1. In the **Monitoring Options** tab, click the **Set up monitoring** link.
2. Select **Set up a new monitoring task**, and click **OK**. The New Monitoring Task Wizard opens.
3. In the Define Monitor dialog box, select **Server Analysis Agents** in the Monitor Type field.
4. In the Choose Counters dialog box, click the Counters tab, then select **Top Process Monitoring** and add it to the Selected Items pane.
5. Complete the steps in the wizard to save the monitoring task.

When your test is finished, the Top Processes data collected will be included in your test's timing file which you can open in QALoad Analyze.

Dial-up virtual users

QALoad's dial-up/dial-down feature 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 ramp-up session before running the test

> Note: Ramp-up is not supported for a machine assignment entry that is using a player group.

Setting Up a Test

Overview of test setup

To set up a load test, you will 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 in the Getting Started section.

Determining general Conductor behavior

General Conductor options you set will be applicable for all tests run until you change them. 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 from the Conductor's Options dialog box.

Setting up a specific test session

To prepare the Conductor for a specific test, you will save information and parameters specific to that test into a reusable session ID file (.id). You will need to 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 *QA*Load 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
- (Optional) configuration for server monitoring
- (Optional) integration with other Compuware products.

All of the above information can be entered and saved from the Conductor's main window, the Test Information Window.

Anticipating error conditions

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

QALoad helps you anticipate error conditions and determine, before running your test, how your Players will 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.

To set the error handling option, see the help topic for the Script Assignment tab.

Configuring the Conductor

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

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

1. From the main menu, choose **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.
3. When you are finished, click **OK** to save your changes. Any options you set will apply to all tests until you change them.

For detailed descriptions of the options that are available, see Options dialog box.

Managing large amounts of test 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.

To thin timing data:

1. With your test session ID file open, click the **Script Assignment** tab.
2. For each script for which you would like to thin your test data, click the button in the **Timing Options** column.
3. On the Timing Options dialog box, click the **Enable Timing Data Thinning** check box.
4. In the **Thin Every...** field, type the number of transactions to average. The average will be sent to the Conductor for inclusion in the timing file, rather than every value.
5. Click **OK**.
6. Save your changes to your test session ID file by choosing **File>Save** from the Conductor menu.

For more details about the Timing Options dialog box, see Timing Options.

Setting up a test session

You can enter all the information necessary for your session ID file right from the Conductor's main window, the Test Information Window.

Hint: The following procedures guide you through setting up a reusable test session ID using the Conductor's main window, the Test Information window. Follow each step in turn to configure your test, or revisit this help topic later to make changes to any specific part of the test setup.

Step 1: Enter descriptive information about the test

On the Test Information tab:

1. (Optional) Type descriptive information about the test in the **Test Description**, **Client System**, **Server System**, **Database Size**, and **Comments** fields.
2. In the **Session Duration** field, type a time limit to specify the maximum duration for the test to run. Enter zero if you do not want to specify a maximum duration.

Hint: For details about any field on the Test Information tab, see Test Information.

Step 2: Assign compiled scripts to the test

On the Script Assignment tab:

1. Click **New** to open the Select Script dialog box. The Select Script dialog box lists the scripts available for your transaction type. If it does not, select your transaction type (middleware environment) from the **Scripts of Type** list.

Hint: To open the Select Script dialog box from the Script column, click in the Script column to enable the Browse (...) button. Then, click Browse.

The Select Script dialog box lists a status for each script that indicates whether the script is compiled. If it is not, you must compile the script before attempting to use it in a test.

2. Select a script from the list and click **Select** to return to the **Script Assignment** tab.

3. Continue selecting scripts until all scripts you wish to use in this test are listed.

4. (Optional) Select **ApplicationVantage Mode** to enable AV timings. When you make this selection. you must also perform Step 5: Set up ApplicationVantage Options.

Hint: For details about any field on the Script Assignment tab, see Script Assignment.

Step 3: Set test options for each script

For each assigned script on the Script Assignment tab:

1. In the **Transactions** column, type the number of transactions that each virtual user running this script should run. Once a 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
2. Click in the **Debug Options** column to enable the **Browse** button. Click **Browse** to open the Debug Options dialog box, and then set any options for **Debug Trace** and **Logfile generation**. For a description of the Debug Options dialog box, see Debug Options.
3. In the **Error Handling** column, select the option that indicates how the Player running this script should behave when encountering non-fatal errors: **Abort the transaction**, **Continue the transaction**, or **Restart the transaction**.
4. Enter a value in the **Sleep Factor** column 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).
5. In the **Service Level Threshold** column, type a maximum duration for this script. At runtime, the *QA*Load Conductor will display a runtime graph comparing the **Service Level Threshold** with the actual duration.
6. In the **Pacing** column, type a value, in seconds, for pacing.
7. Click in the **Timing Options** column to enable the **Browse** button. Then, click **Browse** to access the Timing Options dialog box and set options related to checkpoints and data thinning.
8. (Optional) Click in the **External Data** column to enable the **Browse** button. Then, click **Browse** to open the External Data dialog box and associate any necessary external files with your selected script.

Hint: For details about any field on the Script Assignment tab, see Script Assignment.

Step 4: Set up Player machines

From any tab:

1. Select **Tools>Manage Player Machines**. The **Manage Player Machines and Groups** dialog box appears.
2. Click **File>Discover Player Machines** to query your network for *QA*Load Player workstations. All workstations with *QA*Load Players installed are listed. If Player machines are discovered to have previous versions of *QA*Load installed, an error message informs you which machines need to be updated.
3. Check the availability of all the Player machines on your network by clicking **Verify All**, or by selecting individual machines and clicking **Verify**. The *QA*Load Conductor requests each selected Player machine to ensure it is available.

   If a Player machine is available, system information for that machine appears in the Details area of the dialog box. View the Properties dialog box for that Player machine by double-click on the Player machine listing. If the Player machine is not available, you receive a message that the Player is not responding.
4. (Optional) Use **New** to manually add a Player machine, or **Delete** or **Delete All** to remove machines. To save the current machine setup for re-use, create a new configuration file (.cfg). How?

Hint: For details about any field on the Machine Configuration tab, see Monitoring Options.

Step 5: (Optional) Set up ApplicationVantage

Note: You must have ApplicationVantage installed on the Player machine to activate these fields.
This step is required if you selected ApplicationVantage Mode on the Select Script dialog box . Refer to Step 2: Assign Compiled Scripts to the Test for more information.

From any tab:

1. Select **Tools>Manage Player Machines**. The **Manage Player Machines and Groups** dialog box appears.
2. Double-click on the Player machine listing. The Edit Player Machine dialog box displays.
3. Click the **ApplicationVantage Settings** tab.
4. In the **NIC ServiceName** field, select the Network Interface Card on which you will collect timings.
5. Click **Apply**, then click **OK**.

Step 6: (Optional) Set up Remote Monitor Machines

On the Monitoring Options tab:

1. Click the **Enable Performance Monitoring at Runtime** check box to enable monitoring at test time.
2. Create a monitoring task to specify which counters to monitor or apply an existing task and modify the counters as necessary:
   - Create a new monitoring task
   - Use an existing task

Step 7: Assign scripts to Player machines

On the Machine Assignment tab, the scripts you assigned to the test on the Script Assignment tab are listed in the Script column. Fill in the following columns:

Note: Use Auto Configure to have QALoad automatically assign scripts to virtual users.

1. In the **Starting VUs** column, type the number of virtual users to initially launch the script on this machine when a test begins.
2. In the **VU Increment** column, type the number of virtual users that should be added, at intervals, if you want this machine to add incremental virtual users. You must also fill in the **Time Interval** and **Ending VUs** fields.
3. In the **Time Interval** column, type the time interval at which incremental virtual users should be added to a test. (For example, to add virtual users every 5 minutes, type 00:05:00). You must also fill in the **VU Increment** and **Ending VUs** field.
4. Type the number of virtual users assigned to run until the end of the test.
5. In the **Machine** column for each script, select a Player machine from the drop-down list to assign it to that script. If no Player machines are available in the drop-down list, click the **Manage Player Machines button** to set up a Player.
6. In the **Mode** column, select the test mode for each Player machine: **thread-based** or **process-based**.
7. (Optional) Use **New**, **Delete**, and **Delete All** to add or remove scripts from this test.

Step 8: (Optional - WWW only) Enable Expert User

On the Machine Assignment tab:

1. Click in the **Script** column to enable the **Browse** button. Click **Browse** to open the **Expert User Options** dialog box.
2. Select **Enable Expert User timings**.
3. In the **Virtual User** field, type a virtual user number, if necessary.
4. Click **OK**.

When all scripts have been successfully assigned to Player machines and the test is ready to run, Run on the Machine Assignment tab will become available and you can run a test.

Step 9: Save the test setup you just created as a reusable session ID file

Save the test setup

To save the current test setup to a reusable test file called a session ID, click File>Save to name and save it.

Save the machine configurations

To save the Player Agent, Server Analysis Agent, Remote Monitoring Agent, Expert User, and ApplicationVantage integration configurations to a reusable file, called a configuration file (.cfg) see Saving machine configurations.

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, saving 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 Player Machines and Groups dialog box. The .cfg field always displays the active configuration.

To create a new, empty .cfg file:

1. On the **Monitoring Options** tab, click Tools>Player Machines. The Manage Player Machines and Groups dialog box displays.
2. Click down arrow on the .cfg field at the bottom of the dialog box.
3. Choose **<New>**.
4. 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 Player Machines and 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 Player Machines and Groups** dialog box, click the down arrow on the .cfg field at the bottom of the dialog box.
2. Choose **<Save As>**.
3. On the Save As dialog box, specify a name for the new file and click **Save**.
4. 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 Player Machines and Groups** dialog box, click the down .cfg field at the bottom of the dialog box.
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.

Add a Player workstation to a test session ID

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

To add a Player workstation to a test session ID:

1. On the Conductor's Test Information window, click the Machine Configuration tab.
2. Click the **New** button to open the New Entry dialog box.
3. Type the host name and port number of the new Player in the **Agent Machine Host Name** and **Agent Port** fields. If you do not know a Player's host name, check the Player's window at startup. It displays the workstation's Player name.
4. Click the **Request** button to ensure the machine is available. If the Player is available, the number of thread- and process-based virtual users it supports are listed in the appropriate columns, and the button text changes to **OK**.

   If the 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.

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 **Run** menu, choose **Batch Test**.
2. In the **Session Files** (.id) box, highlight the session you want to add, and click the **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 Batch List on the right side of the dialog box.

Assigning scripts to Player workstations

1. There are two ways to assign scripts to Player workstations from the Test Information Window, Machine Assignment tab:
   - Manual assignment — In the Machine column for the appropriate script, select a machine from the list to assign it to that script.
   - Automatic assignment — Type the total number of virtual users to assign to the test in the Starting VUs column, then click the Auto Configure button. QALoad will automatically assign scripts to each Player workstation.
2. From the **File** menu, choose **Save** to save your changes to the current session ID file, or **Save As** to save them to a new session ID file.

Changing the number of virtual users

You can change the number of virtual users assigned to a script from the Test Information Window, Machine Assignment tab.

To change the number of virtual users:

1. Enter 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 and Ending VUs columns for the appropriate script to determine how many virtual users to add at the interval specified in the Time Interval column.
3. From the **File** menu, choose **Save** to save your changes to the current session ID file, or **Save As** to save them to a new session ID file.

Changing test options

To change test options:

1. To change any of your test options, make your changes on the Test Information, Script Assignment, Monitoring Options, or Machine Assignment tabs.
2. From the **File** menu, choose **Save** to save your changes to the current session ID file, or **Save As** to save them to a new session ID file.

Removing a script from a test

Follow these steps to remove a script.

To remove a script from a test:

1. On the **Test Information** tab, click on the selection box to the left of the script name to highlight the row.
2. Click the **Delete** button to remove the script from the test.
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.

Removing a session from a batch test

1. Select **Run>Batch Test**.
2. Highlight the session to remove in the Batch List and click **Remove**.

Setting Auto Abort

1. Select **Run>Batch Test**.
2. Select **Auto Abort After** and use the slider to set the number of seconds to wait before aborting a test.

Normally, each test runs for the duration set in its respective session ID file. An individual test run is complete when all the virtual users have exited. If the Auto Abort After check box is selected and all the virtual users in a test do not exit within the specified number of seconds, the Conductor automatically aborts the test.

Remove used datapool records after a test

You can remove used datapool records after a test 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 which 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 will mark 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 will be used for your subsequent test.

To use the Strip Datapool function:

1. With the current test's session ID file open in the Conductor, select the Script Assignment tab.
2. Click the **External Data** button. The External Data dialog box appears.
3. In the Central Datapool area, select the **Strip** check box. Click **OK**.
4. At the end of your test, a Strip Datapools prompt will appear asking if you wish to go to the Strip Datapools screen. Click **Yes**.
5. The Strip Data Pool dialog box appears. Click the **Strip** button.
6. When you are finished, click **Done**.

Setting automatic stripping of datapools between batch tests

1. Select **Run>Batch Test**.
2. Add sessions to the Batch List, or load the batch file you wish to run.
3. Select **Automatic Datapool Stripping Between Tests**.

Datapool records used during the test will be removed before starting the next test in the batch.

Note: Only those datapools marked as strippable on the External Data dialog box, accessible from the Test Information Screen Script Assignment tab, will be removed.

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 Batch Test dialog box. After each test is complete, the Conductor delays for the specified amount of time before starting the next test.

Validating a script

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 from the Conductor.

Debugging 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. Three debugging strategies are described below.

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.

To enable the Debug Trace option:

1. On the Conductor's Script Assignment tab, highlight the script you want to monitor.
2. In the Debug Options column, click the browse (**...**) button (note that the button may not be visible until you click in the Debug Options column).
3. On the Debug Options dialog box, 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.
4. Click **OK** to save your changes.
5. From the Conductor's main menu, click **File>Save** to save your test session ID.
6. Run your test as usual. Each virtual user for which you enabled Debug Trace will display messages on its assigned Player workstation indicating which commands are being executed.

Log details from selected virtual users while they are running (Citrix, DB2, ODBC, Oracle, Oracle Forms Server, SAP, Uniface, Winsock, or WWW only)

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.

To enable Logfile Generation:

1. On the Conductor's Script Assignment tab, highlight the script you want to monitor.
2. In the Debug Options column, click the browse (**...**) button (note that it might not be visible until you click it).
3. On the Debug Options dialog box, 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.
4. Click **OK** to save your changes.
5. From the Conductor's main menu, click **File>Save** to save your test session ID.
6. Run your test as usual. Each virtual user for which you enabled Logfile Generation will create a file containing information about their performance. After the test is finished, the Conductor will request all log files from the Players and save them in the directory `\Program Files\Compuware\QALoad\LogFiles` on the workstation where the Conductor is installed. Log files are named <*scriptname*>_<*middleware*>_vu<*AbsoluteVirtualUserNumber*>.<*ext*>, where:
   - <*scriptname*> is the name of the script the virtual user ran
   - <*middleware*> is the name of your middleware application
   - <*AbsoluteVirtualUserNumber*> is the identification number assigned to the virtual user

! <*.ext*> is the file extension, dependent upon which middleware application you are testing. File extensions are listed in the following table:

Middleware	File Extension
Uniface WWW	.cap — A standard log file containing information about all statements executed during a test.
Citrix DB2 ODBC Oracle Forms Server SAP Winsock WWW	.log — A standard log file containing information about all statements executed during a test.

View automatically generated log files from failed virtual users (Oracle and WWW only)

A Player that fails to execute a test script automatically generates a log file (with .rip extension) that details the requests that were played back by that Player. You can view a Player's .rip file from the Conductor's Runtime Window while a test is running, or later in QALoad Analyze. At the end of a test, all .rip files are sent from the Players to the `\QALoad\LogFiles` directory and added to the merged timing file, where you can view them in Analyze.

To view a .rip file while the test is running:

1. In the Runtime Window's tree-view, right-click on a failed Player icon.
2. From the context menu, choose **Display RIP File**.

A new tab opens in the Runtime Window displaying details about the script the Player attempted to run, and which subsequently failed.

To view a .rip file in QALoad Analyze:

At the end of your test, after all test results have been collected, open your resulting timing file in Analyze. Click on the RIP Files group to see a list of all .rip files that were collected during the test.

Running a Test

Running a load test

When you have a load test properly set up, you can start the test by clicking Run from the toolbar or by choosing Run>Start Test from the menu.

Tip: While a test is running, the Conductor's interface changes to provide you with real-time test options.

Running a Series of Tests

You can also run a series of tests — a batch test. A batch test is comprised of multiple session ID files. When you run a batch test, the session files are executed sequentially until all of them are executed. The Conductor allows you to run multiple batch tests without operator intervention.

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 will prompt you after you start the test and will attempt to check out the appropriate number of licenses. We recommend, though, 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 opens.
   - If you are licensed for concurrent licensing (multiple Conductors) the Conductor will query your license server to determine how many licenses are currently available, and return 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 will be unavailable, as you will not need to, or be able to, check out virtual user licenses. All virtual users for which you are licensed will be available to only this Conductor. Click Close to return to your test setup.
2. In the Licensing Operations area, type how many virtual user licenses you want to check out in the **Number of Licenses** field.
3. Click **Check Out**. The licenses will be checked out to your Conductor, and 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 opens.
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 will be made available to other Conductor workstations on the network.

Starting a test

Click Run ▶ or choose Start Test from the Conductor's Run menu.

Note: While a test is running, the toolbar changes to display the Runtime Toolbar buttons.

Starting the Conductor from the command line

To start the Conductor from the command line, type:

```
mpwin32 <sessionID_file_name> /l /e /a /t
```

Note: You must have a complete session file before the test can start. For example, you must specify either a set number of transactions or a test duration. A transaction count or a test duration of 0 is not valid.

Valid startup parameters are:

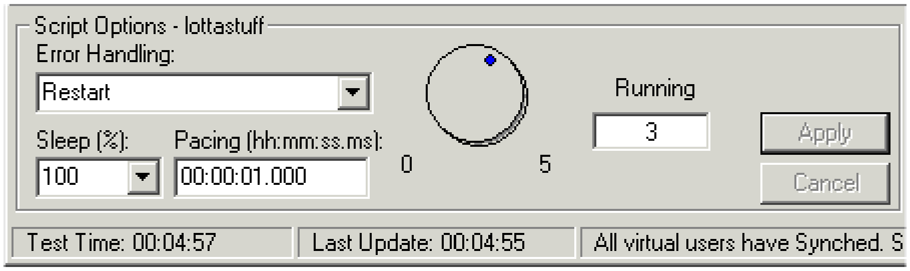
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>/l
(Optional)</td><td>Creates a log file showing error messages and test status.</td></tr>
<tr><td>/e
(Optional)</td><td>Exits the Conductor when the test completes.</td></tr>
<tr><td>/a
(Optional)</td><td>Launches Analyze when the test completes.</td></tr>
<tr><td>/t
(Optional)</td><td>Executes the Conductor at a set time or a set date and time. Time can be specified by either 12-hour or 24-hour format. The following examples of the /t parameter demonstrate each scenario.
<table>
<tr><td>Command</td><td>Result</td></tr>
<tr><td>/t"03:30:00 pm"</td><td>Starts the Conductor at 3:30PM today.</td></tr>
<tr><td>/t06:00:00</td><td>Starts the Conductor at 6:00AM today.</td></tr>
<tr><td>/t"12/01/04 03:30:00 pm"</td><td>Starts the Conductor on December 1, 2004 at 3:30PM.</td></tr>
<tr><td>/t"12/25/04 14:00:00"</td><td>Starts the Conductor on December 25, 2004 at 2:00PM.</td></tr>
</table>
</td></tr>
</table>

Dialing up/down virtual users

Notes:

- If you haven't configured a ramp-up session, you will not be allowed to add or suspend virtual users while the test is running. For information about configuring a ramp-up session, see Configuring a ramp-up session.
- Ramp-up 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 Control Panel, a portion of which is shown below:



Script Options - lottastuff

Error Handling: Restart

Sleep (%): 100

Pacing (hh:mm:ss.ms): 00:00:01.000

Running: 3

Test Time: 00:04:57 | Last Update: 00:04:55 | All virtual users have Synched. S

If you click on a Player or script icon in the test's tree-view, the Runtime Control Panel will indicate 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 Running field (or by using the dial control).

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. The Running column in the top pane shows how many virtual users are currently running on that script or Player.
2. In the Runtime Control Panel, type a new number in the **Running** field or drag the dial control to change the number.
3. When you are done, click the **Apply** button. The Conductor will release or suspend the specified number of virtual users.

Your change do not take effect until you click the Apply button.

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. Decreasing the update interval will reduce the amount of overhead incurred in large load tests due to the communications between the Conductor and large numbers of virtual users.

On the Runtime Control 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.

Any change takes effect immediately, and applies to all scripts in the test.

Manually Aborting a Test

To manually abort a test (stop script execution for all workstations), click on the Abort toolbar button ⚠.

Remove used datapool records after a test

You can remove used datapool records after a test 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 which 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 will mark 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 will be used for your subsequent test.

To use the Strip Datapool function:

1. With the current test's session ID file open in the Conductor, select the Script Assignment tab.
2. Click the **External Data** button. The External Data dialog box appears.
3. In the Central Datapool area, select the **Strip** check box. Click **OK**.

4. At the end of your test, a Strip Datapools prompt will appear asking if you wish to go to the Strip Datapools screen. Click **Yes**.
5. The Strip Data Pool dialog box appears. Click the **Strip** button.
6. When you are finished, click **Done**.

Running a Series of Tests (Batch)

Running a batch test

1. Select **Run>Batch Test**.
2. Add the required session ID files to the Batch List using **Add** or **Load**.
3. Click **Start** to initiate the batch.

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

Terminating a batch test

You can stop a batch of tests the same way you would stop a single session test, by clicking the Abort or Exit toolbar buttons.

Monitoring a Running Test

Watching a script execute

Use the Debug window in the Details view of the runtime Conductor to view the executing script. Note that it is possible that you won't see the execution of every statement, in order to minimize network traffic between the Conductor and the Players. The QALOAD.INI file's debug messages-per-sec parameter determines how frequently the Player sends its script debug status to the Conductor. At its default value of one message per second, the Player can execute several statements without sending a debug message to the Conductor.

To open the Debug window, select a workstation in the global control window and click the Debug toolbar button.

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

Viewing datapool usage

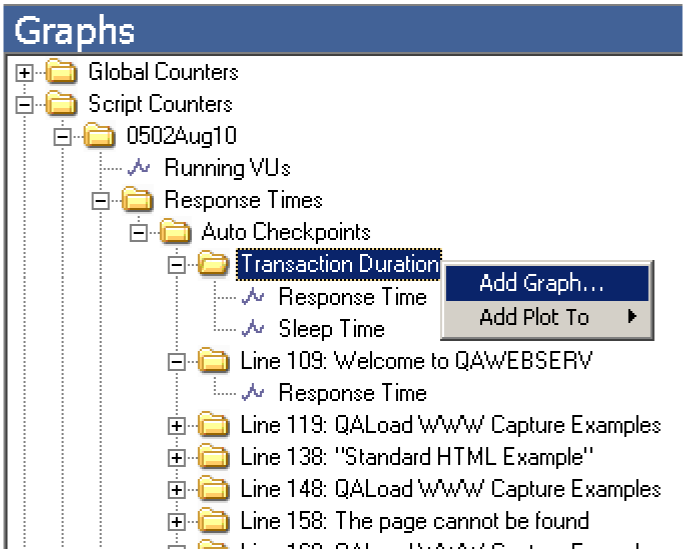
Highlight a script in the global control window and click Datapool to open the Datapool window. As the script executes Get_Data commands, the Datapool window reflects the current datapool record being used by the script.

Graphing checkpoints

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

Selecting checkpoints to graph

Before you can review checkpoint response times in graph form, you must select the checkpoint counters to include. 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.


Graphs
Global Counters
Script Counters
0502Aug10
Running VUs
Response Times
Auto Checkpoints
Transaction Duration
Add Graph...
Add Plot To
Response Time
Sleep Time
Line 109: Welcome to QAWEBSERV
Response Time
Line 119: QALoad WWW Capture Examples
Line 138: "Standard HTML Example"
Line 148: QALoad WWW Capture Examples
Line 158: The page cannot be found

Creating a graph of checkpoint response times

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.

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

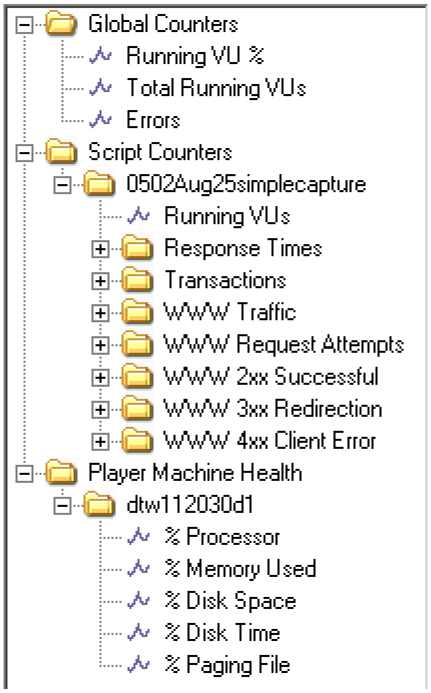
Checkpoint graphs that are created in the Conductor are automatically saved to the current session ID file. To remove all graphs you added, click Graph>Restore Default Graph Layout.

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
  - Running VU %
  - Total Running VUs
  - Errors
- Script Counters
  - 0502Aug25simplecapture
    - Running VUs
    - Response Times
    - Transactions
    - WWW Traffic
    - WWW Request Attempts
    - WWW 2xx Successful
    - WWW 3xx Redirection
    - WWW 4xx Client Error
- Player Machine Health
  - dtw112030d1
    - % Processor
    - % Memory Used
    - % Disk Space
    - % Disk Time
    - % Paging File

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.

Recording and Playing a Test

Recording and replay overview

As a load test is running, the Conductor records each event in a .REC file. After the load test completes, you can open the .REC file to replay the load test and evaluate important events that occurred, such as a sudden increase in processor usage.

When you replay the load test, the playback looks identical to the actual load test, with the exception of the playback toolbar, which appears at the top of the screen. When you open the .REC replay file in the Conductor, you can replay, fast-forward, double fast-forward, or pause.

> Notes: The replay feature provides a visual re-enactment of the load test; it does not perform the actual test or connect to any servers.
> You can only play a .REC file created in the same version as the QALoad Conductor on which it will play. For example, a .REC file created in QALoad 5.02 can only be played using a QALoad Conductor from the 5.02 release.

Recording a test

Load tests can be recorded by selecting the Record Load Test option from the Conductor. When this option is activated and the load test begins, a prompt appears for you to specify a file name for the recording. When the load test is completed, you can replay the test.

To set the Conductor to record a load test:

1. On the **Test Information** tab of the Conductor's main window, select the **Enable Test Recording** check box.
2. Start the load test. The **Record** dialog box appears.
3. Type a name for the record file. Click **OK**.

Replaying a test

If you set the Conductor to record load tests, you can play them back after the test completes. Replaying a recorded load test does not perform the actual load test. A replay provides a visual re-enactment of the events that took place during the load test.

To replay a recorded load test:

1. From the Conductor's **Run** menu, choose **Test Recording>Replay a Load Test**. The **Open Record File** dialog box appears.
2. Browse for the recording file (*filename*`.rec`) that you saved when the load test started. Click **Open**.
3. The test will play back in a viewer that contains a playback toolbar. Use the toolbar buttons described in the following table to control the playback.

Button	Action
⏮	Restarts the test replay from the beginning

▶	Replays the test at normal speed
▶▶	Replays the test twice as fast as normal
▶▶▶	Replays the test four times as fast as normal
❚❚	Pauses the replay at the current snapshot
🚪	Exits test replay and opens the Conductor test setup window

Note: Test control features such as dial-up/dial-down do not work during test replay, but the effects from these features can be observed in the replay. Also, virtual user error details on the Virtual User Info window are not available during replay. Detailed error information is available in the timing file and can be viewed with Analyze.

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 sand 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

You can enable or disable the Expert User for each load test on a script, either before or during the load test.

To enable Expert User before the load test begins:

1. In Conductor, click the **Machine Assignment** tab.
2. Click in the **Middleware** column to enable the **Browse (...)** button. Then, click **Browse**. The **Expert User Options** dialog box displays.
3. Click **Enable Expert User timings**.
4. Type the Virtual User (VU) number to represent the Expert User. The default VU number is zero (0).
5. Click **OK**.

To enable Expert User during the load test:

1. Select the script and any options for the test.
2. From the **Machine Assignment** tab, and click the **Run** button, or click **Run>Start Test**. The Test begins and the Test menus display.
3. Click **Test Options>Expert User Options**. The **Update Expert User Options** dialog box displays listing all the scripts that support Expert User counters.
4. Click the scripts in which you want to enable Expert User, then click **OK**.

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 toolbar button to complete the test and automatically create the timing file (.tim).

Viewing test statistics

Compute test statistics by choosing Launch Analyze from the Conductor's Tools menu or by clicking on the Analyze toolbar button.

Integration and Server Monitoring

Integration and Monitoring Requirements

Integration Requirements

ApplicationVantage

QALoad 5.5 supports integration with ApplicationVantage version 9.8.

ServerVantage

QALoad 5.5 supports integration with ServerVantage 9.7 and 9.8.

Monitoring Requirements

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

Server Analysis Agent Requirements

TCP/IP communications.

The Server Analysis Agent must be installed on the same Workstation/Server as the QALoad Player and therefore shares the same system requirements.

JVM Requirements

SAP, WebLogic, WebSphere, and WebSphere MQ monitoring all require Java Virtual Machine (JVM) installed on the Conductor machine.

- For WebLogic monitoring version 7 and earlier versions, use JVM 1.3. For WebLogic version 8, you must use JVM 1.4 or later. You may also use the JVM that is distributed with the WebLogic Application Server.
- For WebSphere monitoring, use the JVM provided with the WebSphere client or server.
- For WebSphere MQ monitoring, you must use JVM 1.4 or later.
- For SAP monitoring, you must use JVM 1.4 or later.

File Installation Requirements

SAP, WebLogic, WebSphere, and WebSphere MQ monitoring require the following files installed on the Conductor machine.

- For 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.

- For WebLogic monitoring, the ***weblogic.jar*** file must be placed in 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 ***webservices.jar*** file to the same directory. For more information, refer to ***Requirements for WebLogic Remote Monitoring*** in the ServerVantage Reconfigure Agent Online Help.

- 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.
- For WebSphere MQ monitoring, the WebSphere client files listed below must be placed in a directory in 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.
- For 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.
- For 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, SAP, SNMP, 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 Monitoring Options tab.
- 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 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 Setting up a test session.

Using Counters and Templates

About Counters

Using Counters

Counters are the numeric data values that are collected when monitoring servers. For Windows, a large number of performance counters are provided by the operating system registry and Windows server applications. Each resource represents items such as individual processes, sections of shared memory, or physical devices.

Counters exists for components such as processor, memory, processes, hard disk, and cache, with a set of counters that measure statistical information. 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 are cumulative, such as Server - Errors Logon. Some are averages, such as Job Object Details - Page Faults / Sec.

In addition to the numeric value counters, 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.

QALoad's Remote Monitor Agents can monitor the following counter types:

Windows Registry counters

SNMP counters

SAP counters

WebLogic counters

WebSphere counters

WebSphere MQ counters

WMI counters

Instances

When you select a registry counter from the Available Items list in the Choose Counters window, the available instances for that counter appear.

Counters can have none or several instances. For example, if a system has multiple processors, then the counter has multiple instances as well. For counters with multiple instances, a list of the available instances for that counter is presented. Many counters have a _Total instance, which is an aggregate of the individual instances.

For the processor counter object, instances are numbered beginning with 0, on a single processor machine has an instance of _Total and one for 0. If you had a dual-processor machine, you would see 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 are instantaneous, and represent the most recent value for the resource, e.g., Processes, which is the number of processes in the computer at the time of data collection. Others are average values between the last two measurements.

Windows 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

API sockets

Bytes in API notifies

Failed API requests

Failed API sends

GQOS sessions

Interfaces

Network sockets

PATH from API

RESV from API

RSVP msg buffers in use

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

Errors/Sec

Errors During Script Runtime

Requests Succeeded

Requests Timed Out

Requests Total

Errors From ASP Preprocessor

Errors From Script Compilers

Request Bytes In Total

Request Bytes Out Total

Request Execution Time

Request Wait Time

Requests/Sec

Requests Disconnected

Requests Executing

Requests Failed Total

Requests Not Authorized

Requests Not Found

Requests Queued

Requests Rejected

Script Engines Cached

Session Duration

Sessions Current

Sessions Timed Out

Sessions Total

Template Cache Hit Rate

Template Notifications

Templates Cached

Transactions/Sec

Transactions Aborted

Transactions Committed

Transactions Pending

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. This object type handles these registry counters:

Announcements Domain/sec

Announcements Server/sec

Announcements Total/sec

Duplicate Master Announcements

Election Packets/sec

Enumerations Domain/sec

Enumerations Other/sec

Enumerations Server/sec

Enumerations Total/sec

Mailslot Allocations Failed

Mailslot Opens Failed/sec

Mailslot Receives Failed

Mailslot Writes/sec

Mailslot Writes Failed

Missed Mailslot Datagrams

Missed Server Announcements

Missed Server List Requests

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

Async Data Maps/sec

Async Fast Reads/sec

Async MDL Reads/sec

Async Pin Reads/sec

Copy Read Hits %

Copy Reads/sec

Data Flush Pages/sec

Data Flushes/sec

Data Map Hits %

Data Map Pins/sec

Data Maps/sec

Fast Read Not Possibles/sec

Fast Read Resource Misses/sec

Fast Reads/sec

Lazy Write Flushes/sec

Lazy Write Pages/sec

MDL Read Hits %

MDL Reads/sec

Pin Read Hits %

Pin Reads/sec

Read Aheads/sec

Sync Copy Reads/sec

Sync Data Maps/sec

Sync Fast Reads/sec

Sync MDL Reads/sec

Sync Pin Reads/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

Accounting-Responses

Accounting-Responses/sec

Bad Authenticators

Bad Authenticators/sec

Dropped Packets

Dropped Packets/sec

Duplicate Accounting-Requests

Duplicate Accounting-Requests/sec

Malformed Packets/sec

No Record

No Record/sec

Packets Received

Packets Received/sec

Packets Sent

Packets Sent/sec

Unknown Type

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

Accounting-Requests/sec

Accounting-Responses

Accounting-Responses/sec

Bad Authenticators

Bad Authenticators/sec

Dropped Packets

Dropped Packets/sec

Duplicate Accounting-Requests

Duplicate Accounting-Requests/sec

Invalid Requests

Invalid Requests/sec

Malformed Packets

Malformed Packets/sec

No Record

No Record/sec

Packets Received

Packets Received/sec

Packets Sent

Packets Sent/sec

Server Reset Time

Server Up Time

Unknown Type

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

Access-Accepts/sec

Access-Challenges

Access-Challenges/sec

Access-Rejects

Access-Rejects/sec

Access-Requests

Access-Requests/sec

Bad Authenticators

Bad Authenticators/sec

Dropped Packets

Dropped Packets/sec

Duplicate Access-Requests

Duplicate Access-Requests/sec

Malformed Packets

Malformed Packets/sec

Packets Received

Packets Received/sec

Packets Sent

Packets Sent/sec

Unknown Type

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

Access-Accepts/sec

Access-Challenges

Access-Challenges/sec

Access-Rejects

Access-Rejects/sec

Access-Requests

Access-Requests/sec

Bad Authenticators

Duplicate Access-Requests/sec

Invalid Requests

Invalid Requests/sec

Malformed Packets

Malformed Packets/sec

Packets Received

Packets Received/sec

Packets Sent

Packets Sent/sec

Bad Authenticators/sec

Dropped Packets

Dropped Packets/sec

Duplicate Access-Requests

Server Reset Time

Server Up Time

Unknown Type

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

Messages Outbound Errors

Messages Received/sec

Messages Received Errors

Messages Sent/sec

Received Address Mask

Received Address Mask Reply

Received Dest. Unreachable

Received Echo/sec

Received Echo Reply/sec

Received Parameter Problem

Received Redirect/sec

Received Source Quench

Received Time Exceeded

Received Timestamp/sec

Received Timestamp Reply/sec

Sent Address Mask

Sent Address Mask Reply

Sent Destination Unreachable

Sent Echo/sec

Sent Echo Reply/sec

Sent Parameter Problem

Sent Redirect/sec

Sent Source Quench

Sent Time Exceeded

Sent Timestamp/sec

Sent Timestamp Reply/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.

IP Counters

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

Datagrams/sec

Datagrams Forwarded/sec

Datagrams Outbound Discarded

Datagrams Outbound No Route

Datagrams Received/sec

Datagrams Received Address Errors

Datagrams Received Delivered/sec

Datagrams Received Discarded

Datagrams Received Header Errors

Datagrams Received Unknown Protocol

Datagrams Sent/sec

Fragment Re-assembly Failures

Fragmentation Failures

Fragmented Datagrams/sec

Fragments Created/sec

Fragments Re-assembled/sec

Fragments 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.

LogicalDisk Counters

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

% Disk Read Time

% Disk Time

% Disk Write Time

% Free Space

% Idle Time

Avg. Disk Bytes/Read

Avg. Disk Bytes/Transfer

Avg. Disk Bytes/Write

Avg. Disk Queue Length

Avg. Disk Read Queue Length

Avg. Disk sec/Read

Avg. Disk sec/Transfer

Avg. Disk sec/Write

Avg. Disk Write Queue Length

Current Disk Queue Length

Disk Bytes/sec

Disk Read Bytes/sec

Disk Reads/sec

Disk Transfers/sec

Disk Write Bytes/sec

Disk Writes/sec

Free Megabytes

Split IO/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.

Memory Counters

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

% Committed Bytes In Use

Available Bytes

Cache Bytes

Cache Bytes Peak

Cache Faults/sec

Commit Limit

Committed Bytes

Demand Zero Faults/sec

Free System Page Table Entries

Page Faults/sec

Page Reads/sec

Page Writes/sec

Pages/sec

Pages Input/sec

Pages Output/sec

Pool Nonpaged Allocs

Pool Nonpaged Bytes

Pool Paged Allocs

Pool Paged Bytes

Pool Paged Resident Bytes

System Cache Resident Bytes

System Code Resident Bytes

System Code Total Bytes

System Driver Resident Bytes

System Driver Total Bytes

System VLM Commit Charge

System VLM Commit Charge Peak

System VLM Shared Commit Charge

Transition Faults/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 Sent/sec

Bytes Total/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

Bytes Sent/sec

Bytes Total/sec

Current Bandwidth

Output Queue Length

Packets/sec

Packets Outbound Discarded

Packets Outbound Errors

Packets Received/sec

Packets Received Discarded

Packets Received Errors

Packets Received Non-Unicast/sec

Packets Received Unicast/sec

Packets Received Unknown

Packets Sent/sec

Packets Sent Non-Unicast/sec

Packets Sent Unicast/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

Mutexes

Processes

Sections

Semaphores

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

% Disk Time

% Disk Write Time

% Idle Time

Avg. Disk Bytes/Read

Avg. Disk Bytes/Transfer

Avg. Disk Bytes/Write

Avg. Disk Queue Length

Avg. Disk Read Queue Length

Avg. Disk sec/Read

Avg. Disk sec/Transfer

Avg. Disk sec/Write

Avg. Disk Write Queue Length

Current Disk Queue Length

Disk Bytes/sec

Disk Read Bytes/sec

Disk Reads/sec

Disk Transfers/sec

Disk Write Bytes/sec

Disk Writes/sec

Split IO/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.

Print Queue Counters

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

Add Network Printer Calls

Bytes Printed/sec

Enumerate Network Printer Calls

Job Errors

Jobs

Jobs Spooling

Max Jobs Spooling

Max References

Not Ready Errors

Out of Paper Errors

References

Total Jobs Printed

Total Pages Printed

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

Bytes Image Free

Bytes Image Reserved

Bytes Reserved

ID Process

Image Space Exec Read/Write

Image Space Exec Read Only

Image Space Exec Write Copy

Image Space Executable

Image Space No Access

Image Space Read/Write

Image Space Read Only

Image Space Write Copy

Mapped Space Exec Read/Write

Mapped Space Exec Read Only

Mapped Space Exec Write Copy

Mapped Space Executable

Mapped Space No Access

Mapped Space Read/Write

Mapped Space Read Only

Mapped Space Write Copy

Reserved Space Exec Read/Write

Reserved Space Exec Read Only

Reserved Space Exec Write Copy

Reserved Space Executable

Reserved Space No Access

Reserved Space Read/Write

Reserved Space Read Only

Reserved Space Write Copy

Unassigned Space Exec Read/Write

Unassigned Space Exec Read Only

Unassigned Space Exec Write Copy

Unassigned Space Executable

Unassigned Space No Access

Unassigned Space Read/Write

Unassigned Space Read Only

Unassigned Space Write Copy

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

% Processor Time (See Note below.)

% User Time

IO Write Operations/sec

Page Faults/sec

Page File Bytes

Creating Process ID

Elapsed Time

Handle Count

ID Process

IO Data Bytes/sec

IO Data Operations/sec

IO Other Bytes/sec

IO Other Operations/sec

IO Read Bytes/sec

IO Read Operations/sec

IO Write Bytes/sec

Page File Bytes Peak

Pool Nonpaged Bytes

Pool Paged Bytes

Priority Base

Private Bytes

Thread Count

Virtual Bytes

Virtual Bytes Peak

Working Set

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. This object type handles these registry counters:

% DPC Time

% Interrupt Time

% Privileged Time

% Processor Time

% User Time

APC Bypasses/sec

DPC Bypasses/sec

DPC Rate

DPCs Queued/sec

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

Bytes Total/sec

Bytes Transmitted/sec

Connects Core

Connects Lan Manager 2.0

Connects Lan Manager 2.1

Connects Windows NT

Current Commands

File Data Operations/sec

File Read Operations/sec

File Write Operations/sec

Network Errors/sec

Packets/sec

Packets Received/sec

Packets Transmitted/sec

Read Bytes Cache/sec

Read Bytes Network/sec

Read Bytes Non-Paging/sec

Read Bytes Paging/sec

Read Operations Random/sec

Read Packets/sec

Read Packets Small/sec

Reads Denied/sec

Reads Large/sec

Server Disconnects

Server Reconnects

Server Sessions

Server Sessions Hung

Write Bytes Cache/sec

Write Bytes Network/sec

Write Bytes Non-Paging/sec

Write Bytes Paging/sec

Write Operations Random/sec

Write Packets/sec

Write Packets Small/sec

Writes Denied/sec

Writes Large/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

Bytes Received/sec

Bytes Total/sec

Bytes Transmitted/sec

Context Blocks Queued/sec

Logon Total

Pool Nonpaged Bytes

Pool Nonpaged Failures

Pool Nonpaged Peak

Pool Paged Bytes

Errors Access Permissions

Errors Granted Access

Errors Logon

Errors System

File Directory Searches

Files Open

Files Opened Total

Logon/sec

Pool Paged Failures

Pool Paged Peak

Server Sessions

Sessions Errored Out

Sessions Forced Off

Sessions Logged Off

Sessions Timed Out

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

Available Threads

Available Work Items

Borrowed Work Items

Bytes Received/sec

Bytes Sent/sec

Bytes Transferred/sec

Context Blocks Queued/sec

Current Clients

Queue Length

Read Bytes/sec

Read Operations/sec

Total Bytes/sec

Total Operations/sec

Work Item Shortages

Write Bytes/sec

Write 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.

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 deleted

Messages enumerated

Messages in the queue directory

Open message bodies

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

% Recipients Remote

Avg Recipients/msg Received

Avg Recipients/msg Sent

Avg Retries/msg Delivered

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

Local Retry Queue Length

Message Bytes Received/sec

Message Bytes Received Total

Message Bytes Sent/sec

Message Bytes Sent Total

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

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

Inbound Connections Total

Local Queue Length

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

Total messages submitted

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

Alignment Fixups/sec

Context Switches/sec

Exception Dispatches/sec

File Control Bytes/sec

File Control Operations/sec

File Data Operations/sec

File Read Bytes/sec

File Write Bytes/sec

File Write Operations/sec

Floating Emulations/sec

Processes

Processor Queue Length

System Calls/sec

System Up Time

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

Connections Active

Connections Established

Connections Passive

Connections Reset

Segments/sec

Segments Received/sec

Segments Retransmitted/sec

Segments 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.

Telephony Counters

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

Active Lines

Active Telephones

Client Apps

Current Incoming Calls

Current Outgoing Calls

Incoming Calls/sec

Lines

Outgoing Calls/sec

Telephone Devices

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 No Port/sec

Datagrams Received/sec

Datagrams Received Errors

Datagrams 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.

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

Errors During Script Runtime

Errors From ASP Preprocessor

Errors From Script Compilers

Errors/Sec

Memory Allocated

Request Bytes In Total

Request Bytes Out Total

Request Execution Time

Request Wait Time

Requests/Sec

Requests Disconnected

Requests Executing

Requests Failed Total

Requests Not Authorized

Requests Not Found

Requests Queued

Requests Rejected

Requests Succeeded

Requests Timed Out

Requests Total

Script Engines Cached

Session Duration

Sessions Current

Sessions Timed Out

Sessions Total

Template Cache Hit Rate

Template Notifications

Templates Cached

Transactions/Sec

Transactions Aborted

Transactions Committed

Transactions Pending

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

Announcements Server/sec

Announcements Total/sec

Duplicate Master Announcements

Election Packets/sec

Enumerations Domain/sec

Enumerations Other/sec

Enumerations Server/sec

Enumerations Total/sec

Illegal Datagrams/sec

Mailslot Allocations Failed

Mailslot Opens Failed/sec

Mailslot Receives Failed

Mailslot Writes/sec

Mailslot Writes Failed

Missed Mailslot Datagrams

Missed Server Announcements

Missed Server List Requests

Server Announce Allocations Failed/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

Async Data Maps/sec

Async Fast Reads/sec

Async MDL Reads/sec

Async Pin Reads/sec

Copy Read Hits %

Copy Reads/sec

Data Flush Pages/sec

Data Flushes/sec

Data Map Hits %

Data Map Pins/sec

Data Maps/sec

Fast Reads/sec

Lazy Write Flushes/sec

Lazy Write Pages/sec

MDL Read Hits %

MDL Reads/sec

Pin Read Hits %

Pin Reads/sec

Read Aheads/sec

Sync Copy Reads/sec

Sync Data Maps/sec

Sync Fast Reads/sec

Sync MDL Reads/sec

Fast Read Not Possibles/sec

Fast Read Resource Misses/sec

Sync Pin Reads/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)

Filter speed (MBytes/hr)

Total 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

Files to be filtered

Index size (MBytes)

Merge progress

Persistent indexes

Running queries

Total # documents

Unique keys

Wordlists

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

Messages Outbound Errors

Messages Received/sec

Received Timestamp/sec

Received Timestamp Reply/sec

Sent Address Mask

Messages Received Errors

Messages Sent/sec

Received Address Mask

Received Address Mask Reply

Received Dest. Unreachable

Received Echo/sec

Received Echo Reply/sec

Received Parameter Problem

Received Redirect/sec

Received Source Quench

Received Time Exceeded

Sent Address Mask Reply

Sent Destination Unreachable

Sent Echo/sec

Sent Echo Reply/sec

Sent Parameter Problem

Sent Redirect/sec

Sent Source Quench

Sent Time Exceeded

Sent Timestamp/sec

Sent Timestamp Reply/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.

IP Counters

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

Datagrams/sec

Datagrams Forwarded/sec

Datagrams Outbound Discarded

Datagrams Outbound No Route

Datagrams Received/sec

Datagrams Received Address Errors

Datagrams Received Delivered/sec

Datagrams Received Discarded

Datagrams Received Header Errors

Datagrams Received Unknown Protocol

Datagrams Sent/sec

Fragment Re-assembly Failures

Fragmentation Failures

Fragmented Datagrams/sec

Fragments Created/sec

Fragments Re-assembled/sec

Fragments 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.

LogicalDisk Counters

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

% Disk Read Time

% Disk Time

% Disk Write Time

% Free Space

Avg. Disk Bytes/Read

Avg. Disk Bytes/Transfer

Avg. Disk Bytes/Write

Avg. Disk Queue Length

Avg. Disk Read Queue Length

Avg. Disk sec/Read

Avg. Disk sec/Transfer

Avg. Disk sec/Write

Avg. Disk Write Queue Length

Current Disk Queue Length

Disk Bytes/sec

Disk Read Bytes/sec

Disk Reads/sec

Disk Transfers/sec

Disk Write Bytes/sec

Disk Writes/sec

Free Megabytes

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

Available Bytes

Cache Bytes

Cache Bytes Peak

Cache Faults/sec

Commit Limit

Committed Bytes

Demand Zero Faults/sec

Free System Page Table Entries

Page Faults/sec

Page Reads/sec

Pages Output/sec

Pool Nonpaged Allocs

Pool Nonpaged Bytes

Pool Paged Allocs

Pool Paged Bytes

Pool Paged Resident Bytes

System Cache Resident Bytes

System Code Resident Bytes

System Code Total Bytes

System Driver Resident Bytes

System Driver Total Bytes

Page Writes/sec

Pages/sec

Pages Input/sec

Transition Faults/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.

Network Interface Counters

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

Bytes Received/sec

Bytes Sent/sec

Bytes Total/sec

Current Bandwidth

Output Queue Length

Packets/sec

Packets Outbound Discarded

Packets Outbound Errors

Packets Received/sec

Packets Received Discarded

Packets Received Errors

Packets Received Non-Unicast/sec

Packets Received Unicast/sec

Packets Received Unknown

Packets Sent/sec

Packets Sent Non-Unicast/sec

Packets Sent Unicast/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

Mutexes

Processes

Sections

Semaphores

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

% Disk Time

% Disk Write Time

Avg. Disk Bytes/Read

Avg. Disk Bytes/Transfer

Avg. Disk Bytes/Write

Avg. Disk Queue Length

Avg. Disk Read Queue Length

Avg. Disk sec/Read

Avg. Disk sec/Transfer

Avg. Disk sec/Write

Avg. Disk Write Queue Length

Current Disk Queue Length

Disk Bytes/sec

Disk Read Bytes/sec

Disk Reads/sec

Disk Transfers/sec

Disk Write Bytes/sec

Disk Writes/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.

Process Address Space Counters

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

Bytes Free

Bytes Image Free

Bytes Image Reserved

Bytes Reserved

Mapped Space Read Only

Mapped Space Write Copy

Reserved Space Exec Read/Write

Reserved Space Exec Read Only

ID Process

Image Space Exec Read/Write

Image Space Exec Read Only

Image Space Exec Write Copy

Image Space Executable

Image Space No Access

Image Space Read/Write

Image Space Read Only

Image Space Write Copy

Mapped Space Exec Read/Write

Mapped Space Exec Read Only

Mapped Space Exec Write Copy

Mapped Space Executable

Mapped Space No Access

Mapped Space Read/Write

Reserved Space Exec Write Copy

Reserved Space Executable

Reserved Space No Access

Reserved Space Read/Write

Reserved Space Read Only

Reserved Space Write Copy

Unassigned Space Exec Read/Write

Unassigned Space Exec Read Only

Unassigned Space Exec Write Copy

Unassigned Space Executable

Unassigned Space No Access

Unassigned Space Read/Write

Unassigned Space Read Only

Unassigned Space Write Copy

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

% Processor Time (See Note below)

% User Time

Elapsed Time

Handle Count

ID Process

Page Faults/sec

Page File Bytes

Page File Bytes Peak

Pool Nonpaged Bytes

Pool Paged Bytes

Priority Base

Private Bytes

Thread Count

Virtual Bytes

Virtual Bytes Peak

Working Set

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

% Interrupt Time

% Privileged Time

% Processor Time

% User Time

APC Bypasses/sec

DPC Bypasses/sec

DPC Rate

DPCs Queued/sec

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

Bytes Total/sec

Bytes Transmitted/sec

Connects Core

Connects Lan Manager 2.0

Connects Lan Manager 2.1

Connects Windows NT

Current Commands

File Data Operations/sec

File Read Operations/sec

File Write Operations/sec

Network Errors/sec

Read Operations Random/sec

Read Packets/sec

Read Packets Small/sec

Reads Denied/sec

Reads Large/sec

Server Disconnects

Server Reconnects

Server Sessions

Server Sessions Hung

Write Bytes Cache/sec

Write Bytes Network/sec

Write Bytes Non-Paging/sec

Packets/sec

Packets Received/sec

Packets Transmitted/sec

Read Bytes Cache/sec

Read Bytes Network/sec

Read Bytes Non-Paging/sec

Read Bytes Paging/sec

Write Bytes Paging/sec

Write Operations Random/sec

Write Packets/sec

Write Packets Small/sec

Writes Denied/sec

Writes Large/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

Bytes Received/sec

Bytes Total/sec

Bytes Transmitted/sec

Context Blocks Queued/sec

Errors Access Permissions

Errors Granted Access

Errors Logon

Errors System

File Directory Searches

Files Open

Files Opened Total

Logon/sec

Logon Total

Pool Nonpaged Bytes

Pool Nonpaged Failures

Pool Nonpaged Peak

Pool Paged Bytes

Pool Paged Failures

Pool Paged Peak

Server Sessions

Sessions Errored Out

Sessions Forced Off

Sessions Logged Off

Sessions Timed Out

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

Available Threads

Available Work Items

Borrowed Work Items

Bytes Received/sec

Bytes Sent/sec

Bytes Transferred/sec

Context Blocks Queued/sec

Current Clients

Queue Length

Read Bytes/sec

Read Operations/sec

Total Bytes/sec

Total Operations/sec

Work Item Shortages

Write Bytes/sec

Write 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.

SMTP Server Counters

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

% Recipients Local

% Recipients Remote

Avg Recipients/msg Received

Avg Recipients/msg Sent

Avg Retries/msg Delivered

Avg Retries/msg Sent

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

Message Bytes Received/sec

Message Bytes Received Total

Message Bytes Sent/sec

Message Bytes Sent Total

Message Bytes Total

Message Bytes Total/sec

Message Delivery Retries

Message Received/sec

Message Send Retries

Messages Delivered/sec

Messages Delivered Total

Messages Received Total

Bytes Received/sec

Bytes Received Total

Bytes Sent/sec

Bytes Sent Total

Bytes Total

Bytes Total/sec

Connection Errors/sec

Directory Drops/sec

Directory Drops Total

Directory Pickup Queue Length

DNS Queries/sec

DNS Queries Total

ETRN Messages/sec

ETRN Messages Total

Inbound Connections Current

Inbound Connections Total

Local Queue Length

Local Retry Queue Length

Messages Refused for Address Objects

Messages Refused for Mail Objects

Messages Refused for Size

Messages Retrieved/sec

Messages Retrieved Total

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

Remote Queue Length

Remote Retry Queue Length

Routing Table Lookups/sec

Routing Table Lookups Total

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

% Total DPC Time

% Total Interrupt Time

% Total Privileged Time

% Total Processor Time

% Total User Time

File Read Operations/sec

File Write Bytes/sec

File Write Operations/sec

Floating Emulations/sec

Processor Queue Length

System Calls/sec

Alignment Fixups/sec

Context Switches/sec

Exception Dispatches/sec

File Control Bytes/sec

File Control Operations/sec

File Data Operations/sec

File Read Bytes/sec

System Up Time

Total APC Bypasses/sec

Total DPC Bypasses/sec

Total DPC Rate

Total DPCs Queued/sec

Total 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.

TCP Counters

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

Connection Failures

Connections Active

Connections Established

Connections Passive

Connections Reset

Segments/sec

Segments Received/sec

Segments Retransmitted/sec

Segments 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.

Telephony Counters

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

Active Lines

Active Telephones

Client Apps

Current Incoming Calls

Current Outgoing Calls

Incoming Calls/sec

Lines

Outgoing Calls/sec

Telephone Devices

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

% Processor Time

% User Time

Context Switches/sec

Elapsed Time

ID Process

ID Thread

Priority Base

Priority Current

Start Address

Thread State

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 No Port/sec

Datagrams Received/sec

Datagrams Received Errors

Datagrams 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.

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

Active Users

Alerts

Page/Roll Area

Page/Roll Area Max

Process Monitoring

- Buffer Statistics
- CCMS Monitoring
- Connection Test (SM59)
- CPU Consumption
- Itemized Active Users
- Itemized Job Status
- Itemized Spool Queue
- Job Status
- Memory Usage
- Number of Dumps
- Spool Queue
- System Log Entries
- 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.

Data 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 1 import, 1 export, and 1 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.

See User Function Call Example.

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)

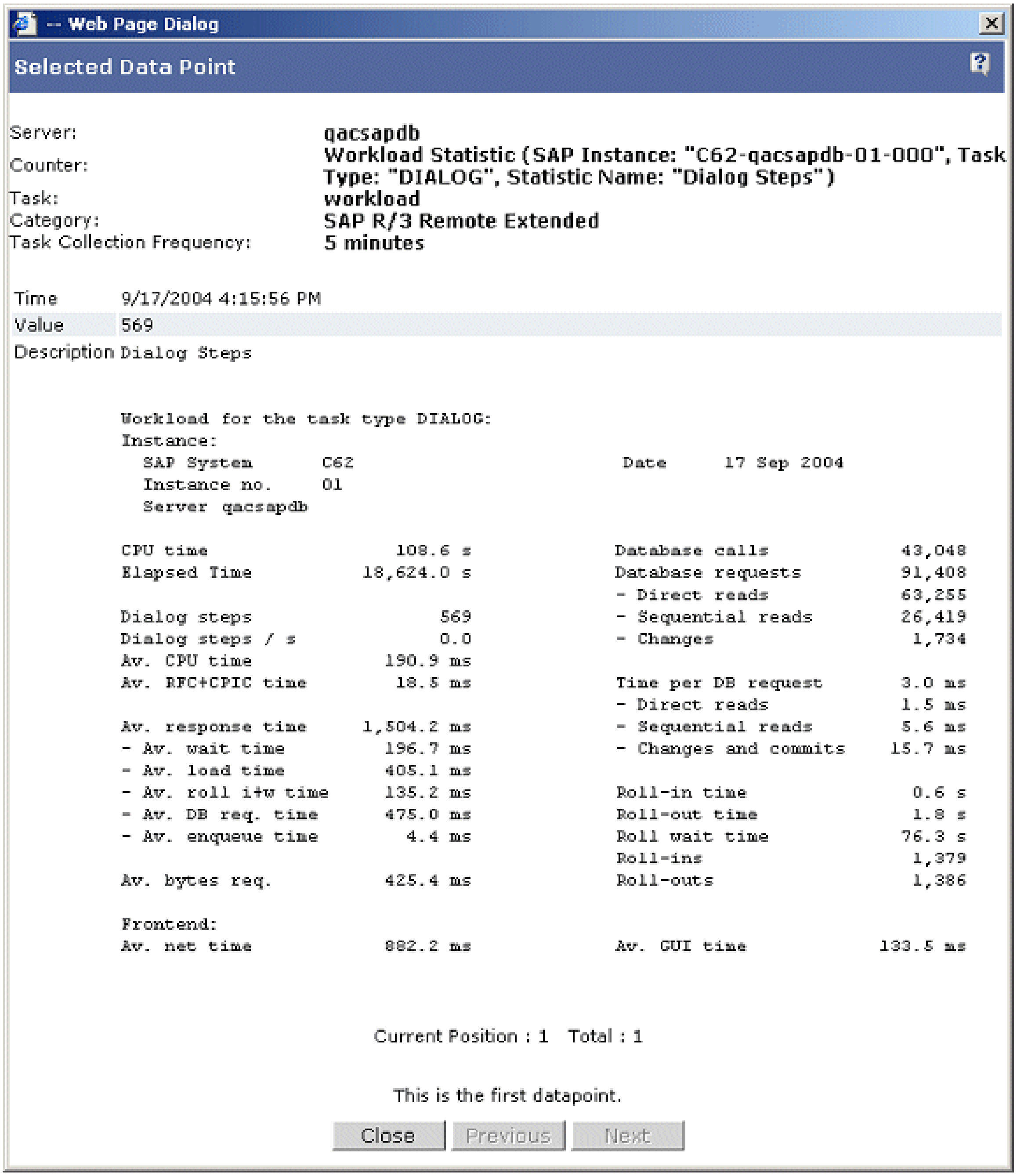
Data Point

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. 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. 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.

WebLogic 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 Connector Service Runtime

WebLogic Deployer Runtime

WebLogic Domain Log Handler Runtime

WebLogic Domain Runtime

WebLogic EJB Cache 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 JMS Session Runtime

WebLogic JTA Recovery Runtime

WebLogic JTA Runtime

WebLogic JVM Runtime

WebLogic Log Broadcaster Runtime

WebLogic Message Driven EJB 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 6.x	WL 7.x	WL 8.x

CachingDisabled	Private property that disables caching in proxies.	Boolean	No	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered	Boolean	No	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.

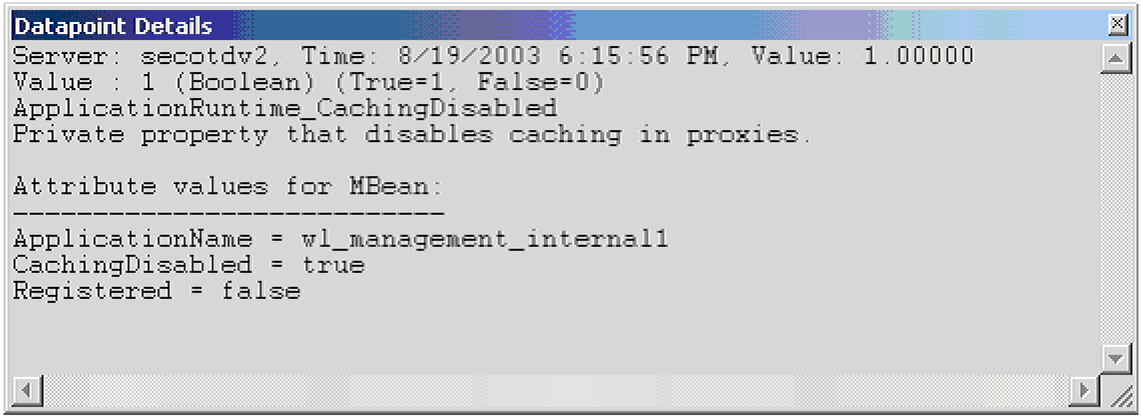
Data Point

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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	Yes
ConnectionPoolCurrentCount	Returns the number of currently deployed connection pools.	Integer	Yes	Yes	Yes
ConnectionPoolsTotalCount	Returns the total number of deployed connection pools instantiated since the Server startup.	Integer	Yes	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	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.

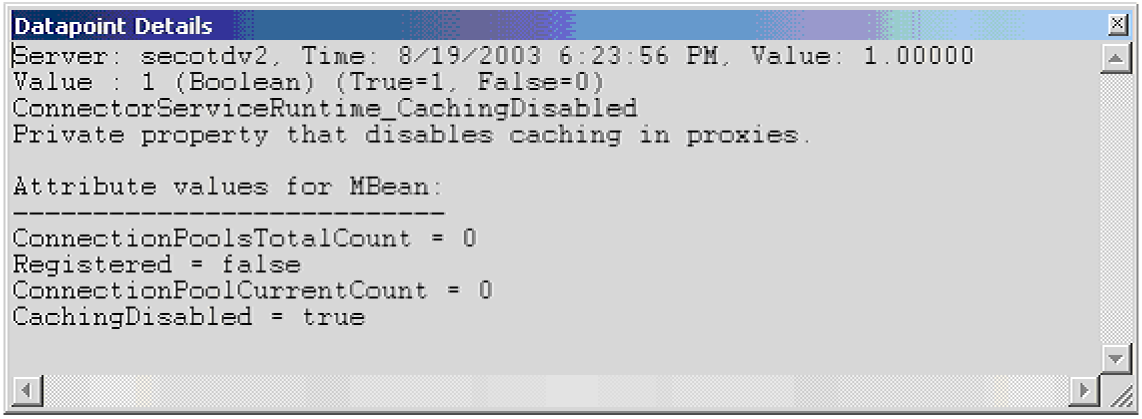
Data Point

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.



Datapoint Details

```
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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	No	Yes	Yes

Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	No	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.

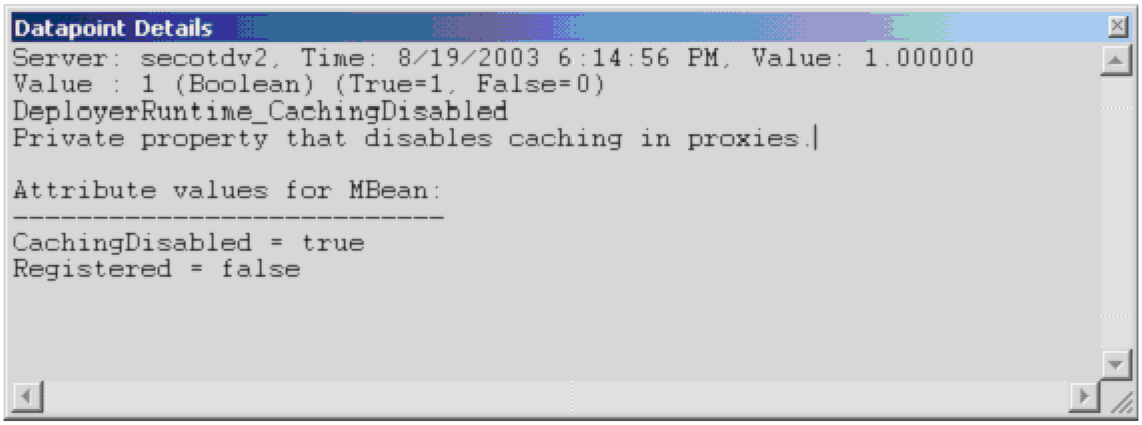
Data Point

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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	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

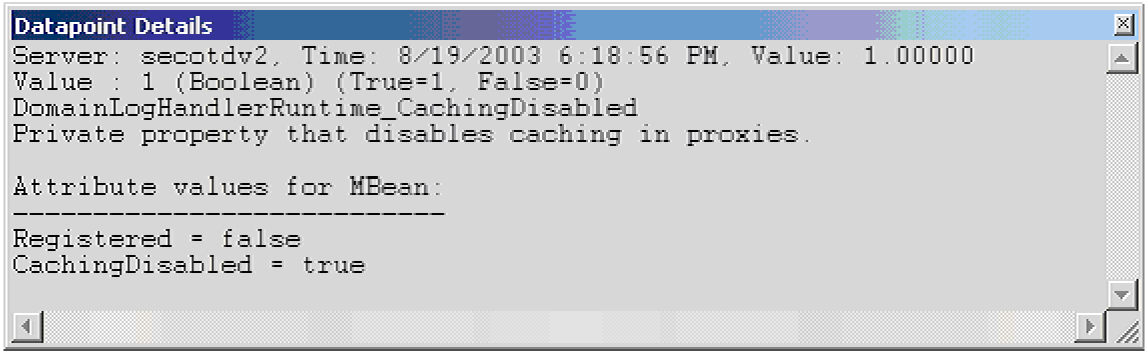
The name of the domain log handler to be monitored. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	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	No	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	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

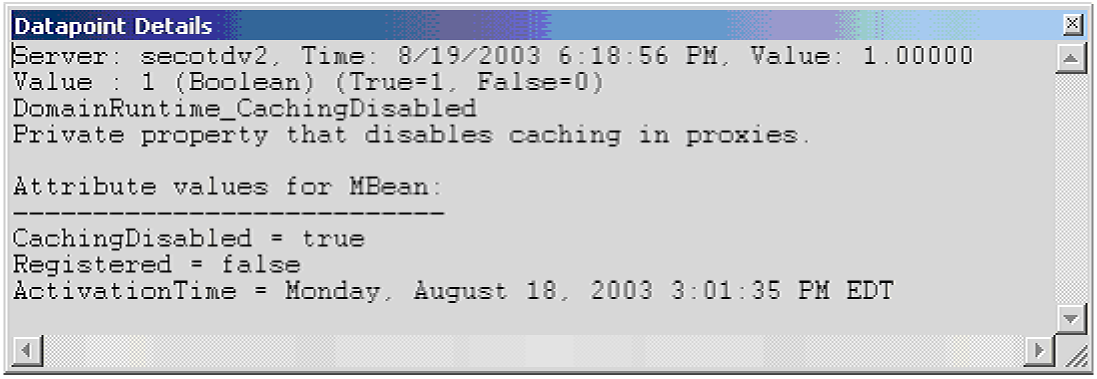
WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

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 6.x	WL 7.x	WL 8.x
ActivationCount	Returns the total number of times the EJB was activated.	Long	Yes	Yes	Yes
CacheAccessCount	Returns the total number of times EJB was accessed in the cache.	Long	Yes	Yes	Yes
CachedBeansCurrentCount	Returns the current number of cached	Integer	Yes	Yes	Yes

	EJBs.				
CacheHitCount	Returns the total number of times the EJB was hit in the cache.	Long	Yes	Yes	Yes
CacheMissCount	Returns the total number of times an attempt to access a bean from the cache failed.	Long	No	No	Yes
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	Yes
PassivationCount	Returns the total number of times the EJB was passivated.	Long	Yes	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	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.

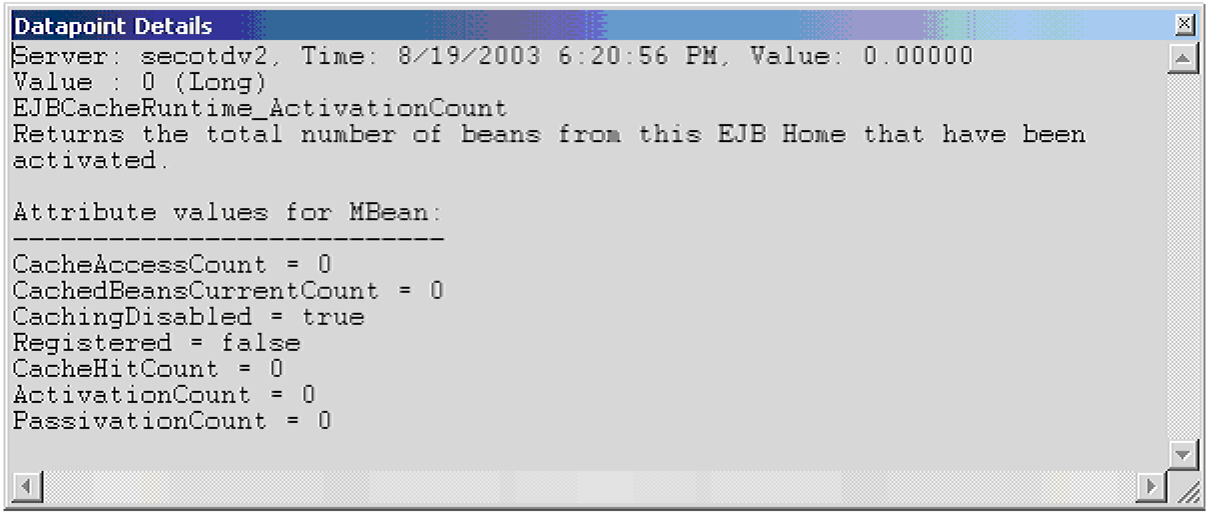
Data Point

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.



```
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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	Yes
DeploymentState	Returns current deployment state of the module.	Integer	No	No	Yes

Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	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	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.

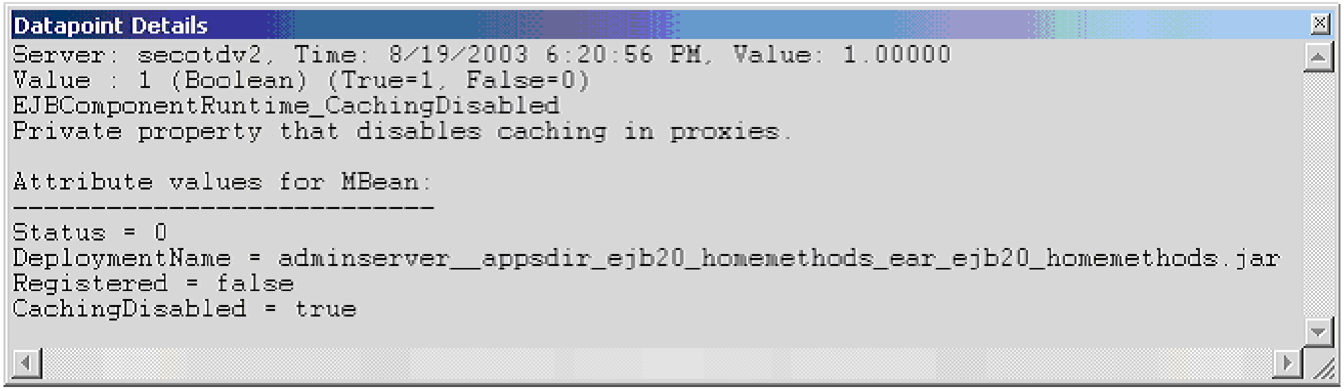
Data Point

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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	No
LockEntriesCurrentCount	Returns the number of current EJB lock entries.	Integer	Yes	Yes	No
LockManagerAccessCount	Returns the number of accesses to the lock manager.	Long	Yes	Yes	No
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes	No
TimeoutTotalCount	Returns the number of objects timed out while waiting on the lock.	Long	Yes	Yes	No
WaiterTotalCount	Returns the number of objects waiting on the lock.	Long	Yes	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:

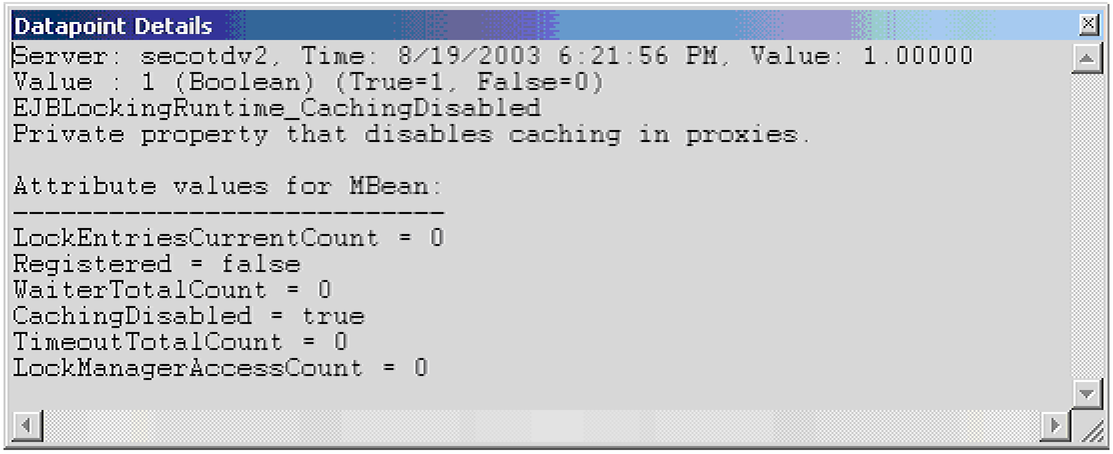
ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

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 6.x	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	No	Yes
BeansInUseCount	Returns the number of beans currently in use.	Integer	Yes	Yes	Yes
BeansInUseCurrentCount	Returns the number of bean instances currently in use from the free pool.	Integer	No	No	Yes
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	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	No	Yes
IdleBeansCount	Returns the number of idle beans in this EJB.	Integer	Yes	Yes	Yes

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	No	No	Yes
PooledBeansCurrentCount	Returns the current number of available bean instances in the free pool.	Integer	No	No	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes	Yes
TimeoutTotalCount	Returns the total number of timed out transactions.	Long	Yes	Yes	Yes
WaiterCurrentCount	Returns the current number of available bean instances in the free pool.	Integer	No	No	Yes
WaiterTotalCount	Returns the number of EJBs currently waiting.	Long	Yes	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:

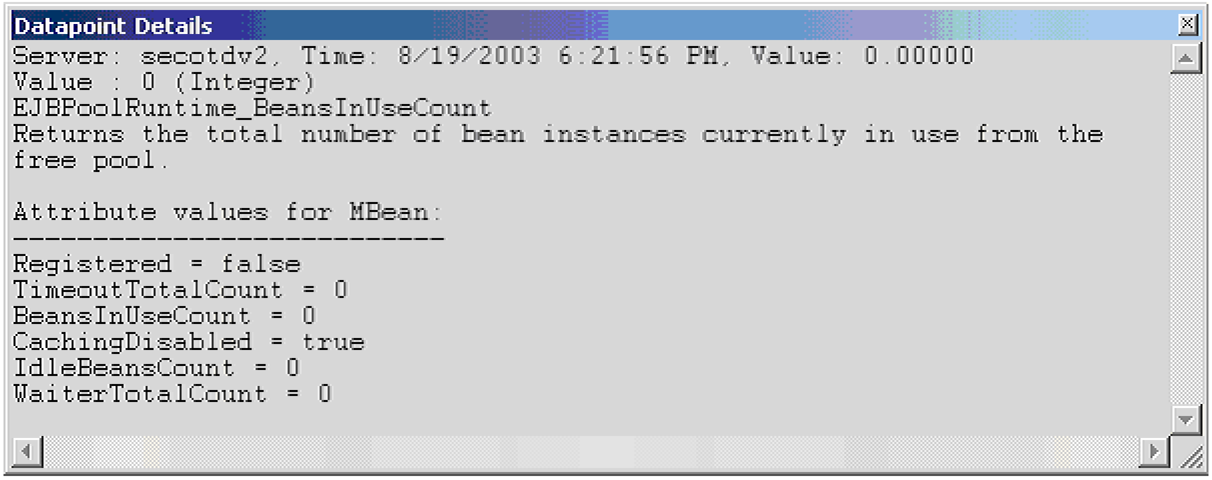
ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes	Yes
TransactionsCommittedTotalCount	Returns the total number of EJB transactions that were committed.	Long	Yes	Yes	Yes
TransactionsRolledBackTotalCount	Returns the total number of EJB transactions rolled back.	Long	Yes	Yes	Yes
TransactionsTimedOutTotalCount	Returns the total number of EJB transactions that timed out.	Long	Yes	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.

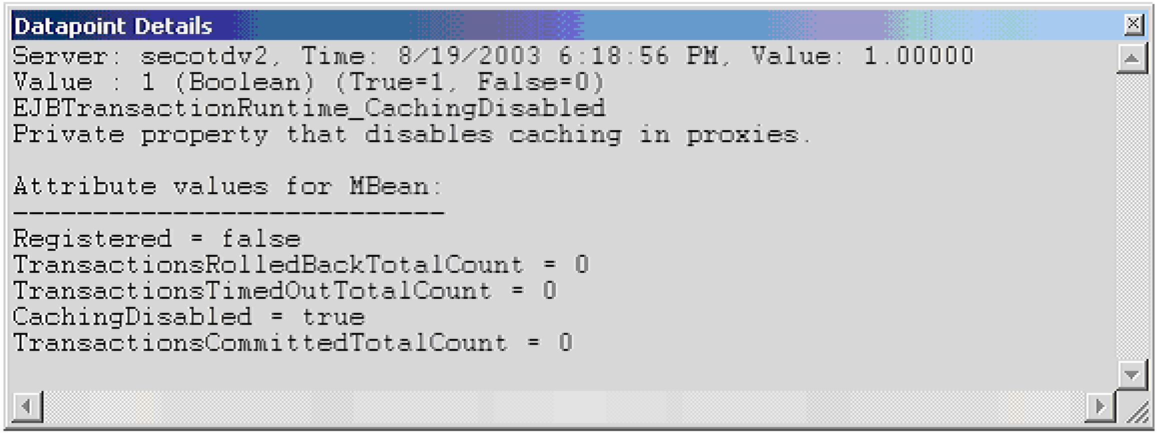
INTERVAL The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

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.



```
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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	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

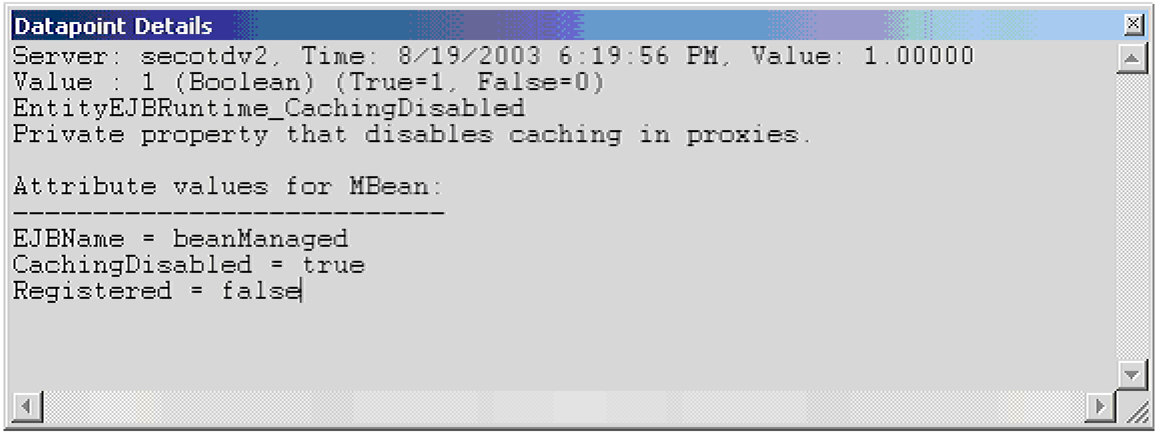
The remainder of the EJB name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	Yes
ExecuteThreadCurrentIdleCount	Returns the number of idle threads assigned to the queue.	Integer	Yes	Yes	Yes
ExecuteThreadTotalCount	Returns the total number of execute threads assigned to the queue.	Integer	No	No	Yes
PendingRequestCurrentCount	Returns the number of waiting requests in the queue.	Integer	Yes	Yes	Yes
PendingRequestOldestTime	Returns the time that the longest waiting request was placed in the queue.	Long	Yes	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes	Yes
ServicedRequestTotalCount	Returns the number of requests that have been processed by this queue.	Integer	Yes	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:

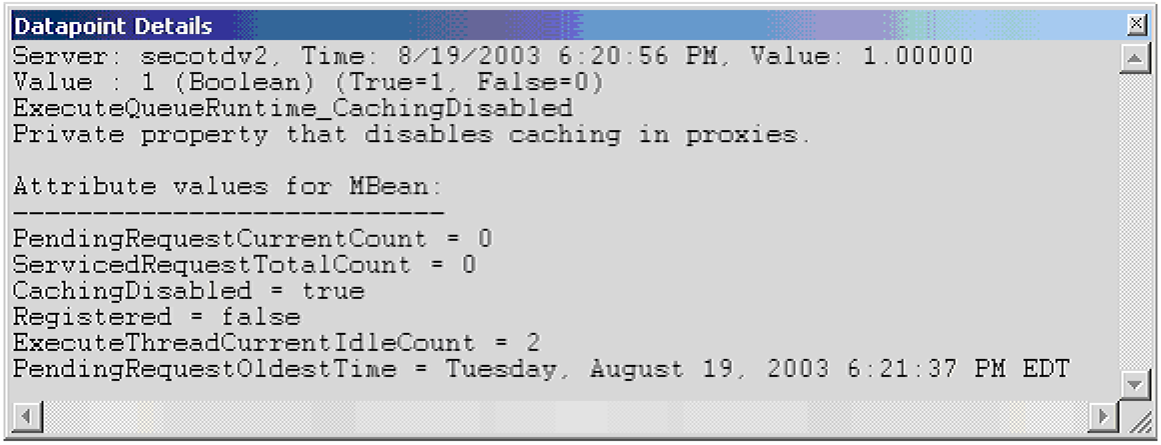
ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

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 6.x	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	No	Yes
ActiveConnectionsCurrentCount	Returns the current number of active connections.	Integer	Yes	Yes	Yes
ActiveConnectionsHighCount	Returns the highest number of active current connections. The count starts at zero each time the JDBCConnectionPoolRuntime MBean is instantiated.	Integer	Yes	Yes	Yes
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	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,	Integer	Yes	Yes	Yes

	divided by summary number of connections.				
ConnectionLeakProfileCount	Returns the current number of connection leak profiles in the profile storage.	Integer	Yes	Yes	Yes
ConnectionsTotalCount	Returns the total number of JDBC connections in this JDBCConnectionPoolRuntimeMBean since the pool was instantiated.	Integer	Yes	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	Yes
HighestNumAvailable	Returns the highest number of available connections in this pool.	Integer	No	No	Yes
HighestNumUnavailable	Returns the highest number of unavailable connections in this pool.	Integer	No	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	Yes
MaxCapacity	Returns the maximum capacity of this connection pool.	Integer	Yes	Yes	Yes
NumAvailable	Returns the number of available connections in this pool.	Integer	No	No	Yes
NumUnavailable	Returns the number of unavailable connections in this pool.	Integer	No	No	Yes
PoolState	Returns true if the pool is enabled, false if the pool is disabled.	Boolean	Yes	Yes	Yes
PreparedStatementCacheProfileCount	Returns the number of prepared statement cache profiling stores cache snapshots that are in external	Integer	Yes	Yes	Yes

	storage.				
PrepStmtCacheHitCount	Returns the cumulative, running count of the use of each cached statement.	Integer	Yes	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	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes	Yes
StatementProfileCount	Returns the number of statement profiling stores in external storage.	Integer	Yes	Yes	Yes
WaitingForConnectionCurrentCount	Returns the current number of requests waiting for a connection.	Integer	Yes	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	Yes
WaitSecondsHighCount	Returns the highest number of seconds a connection waited.	Integer	Yes	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.

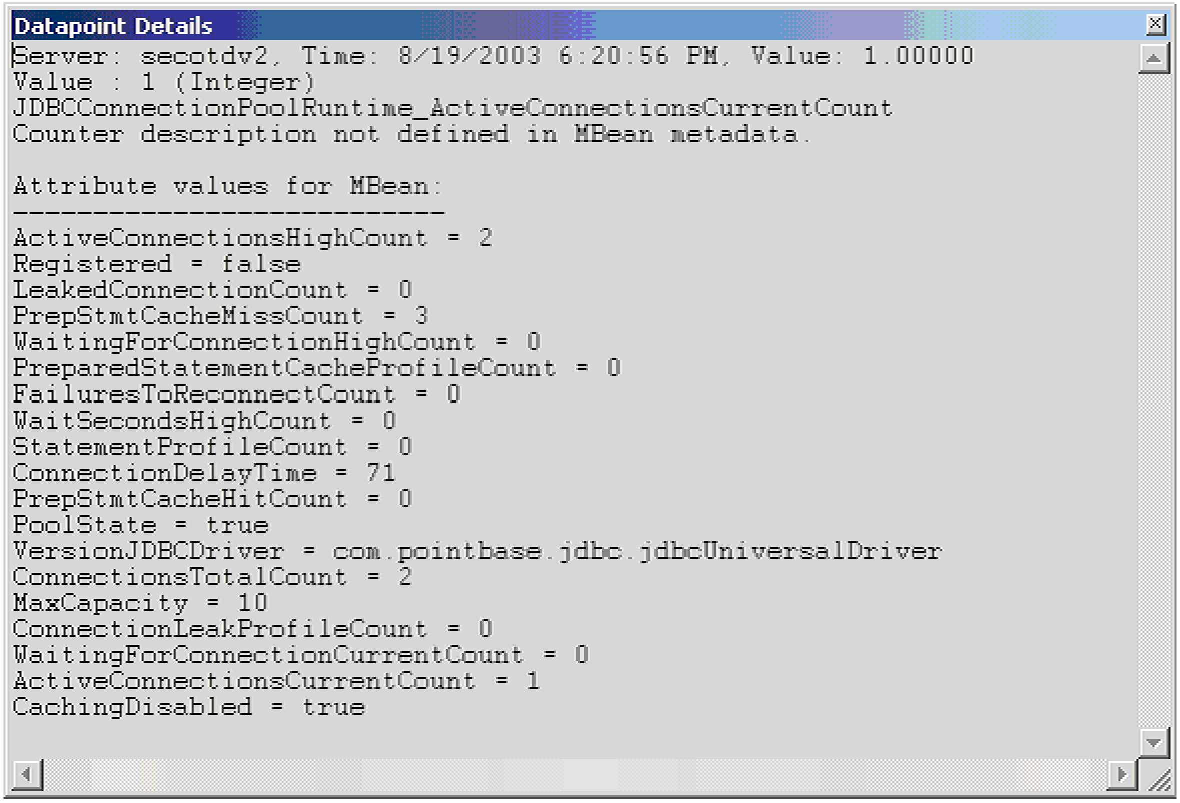
Data Point

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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	No	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	No	Yes	Yes
SessionsCurrentCount	Returns the current number of sessions for this connection.	Long	No	Yes	Yes
SessionsHighCount	Returns the peak number of sessions for this connection since the last reset.	Long	No	Yes	Yes
SessionsTotalCount	Returns the number of sessions on this connection since the last reset.	Long	No	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.

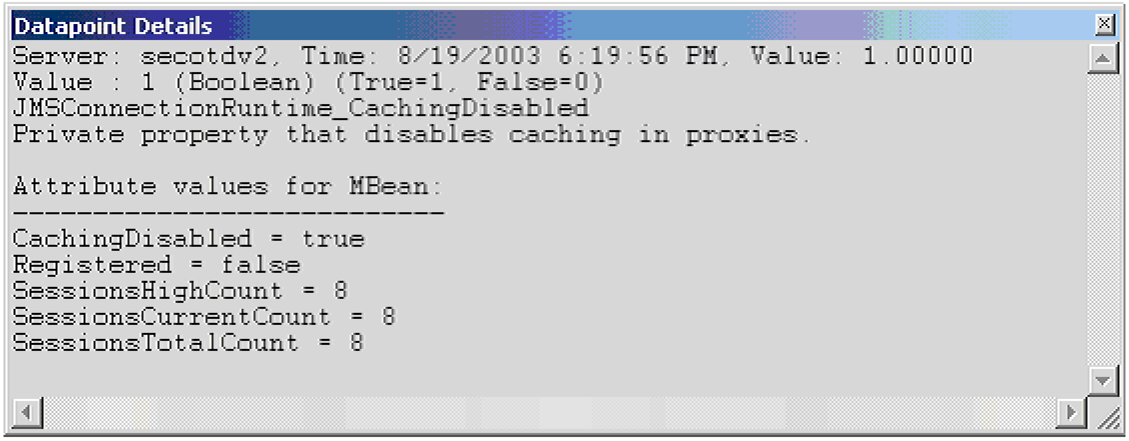
INTERVAL The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

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 6.x	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	No	Yes	Yes

BytesPendingCount	Returns the number of bytes pending (uncommitted and unacknowledged) by this consumer.	Long	No	Yes	Yes
BytesReceivedCount	Returns the number of bytes received by this consumer since the last reset.	Long	No	Yes	Yes
CachingDisabled	Private property that disables caching in proxies.	Boolean	No	Yes	Yes
Durable	Determines whether the consumer is durable.	Boolean	No	Yes	Yes
MessagesPendingCount	Returns the number of messages pending (uncommitted and unacknowledged) by this consumer.	Long	No	Yes	Yes
MessagesReceivedCount	Returns the number of messages received by this consumer since the last reset.	Long	No	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	No	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.

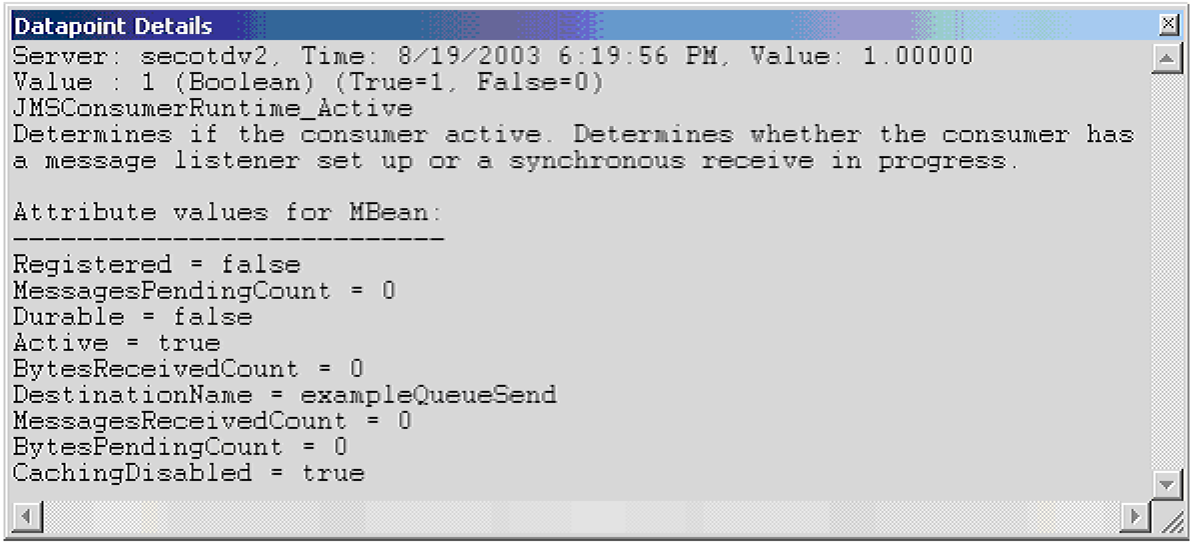
INTERVAL The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

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 6.x	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	No	Yes	Yes
BytesHighCount	Returns the peak number of bytes stored in the destination since the last reset.	Long	No	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	No	Yes	Yes
BytesReceivedCount	Returns the number of bytes received in this destination since the last reset.	Long	No	Yes	Yes
BytesThresholdTime	Returns the amount of time in the threshold condition since the last reset.	Long	No	Yes	Yes
CachingDisabled	Private property that disables caching in proxies.	Boolean	No	Yes	Yes
ConsumersCurrentCount	Returns the current number of consumers accessing this destination.	Long	No	Yes	Yes
ConsumersHighCount	Returns the peak number of consumers accessing this destination since the last reset.	Long	No	Yes	Yes
ConsumersTotalCount	Returns the total number of consumers accessing this destination since the last reset.	Long	No	Yes	Yes
MessagesCurrentCount	Returns the current number of messages in the destination. This does not include the pending messages.	Long	No	Yes	Yes
MessagesHighCount	Returns the peak number of messages in the destination since the last reset.	Long	No	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	No	Yes	Yes
MessagesReceivedCount	Returns the number of messages received in this destination since that reset.	Long	No	Yes	Yes
MessagesThresholdTime	Returns the amount of time in the threshold condition since the last reset.	Long	No	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	No	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:

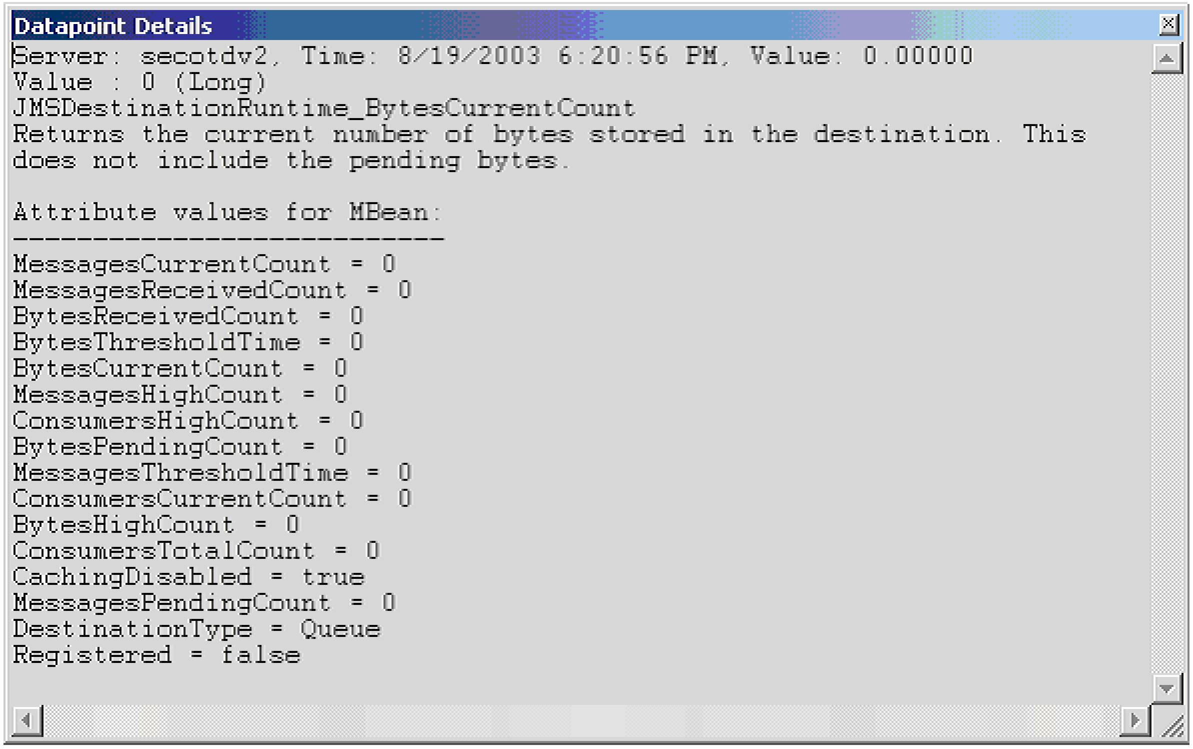
ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	Yes
ConnectionsCurrentCount	Returns the current number of connections to this WebLogic Server.	Long	Yes	Yes	Yes
ConnectionsHighCount	Returns the peak number of connections to this WebLogic Server since the last reset.	Long	Yes	Yes	Yes
ConnectionsTotalCount	Returns the total number of connections made to this WebLogic Server since the last reset.	Long	Yes	Yes	Yes

JMSServersCurrentCount	Returns the current number of JMS servers that are deployed on this WebLogic Server instance.	Long	Yes	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	Yes
JMSServersTotalCount	Returns the number of JMS servers that were deployed on this WebLogic Server instance since the server was started.	Long	Yes	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	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.

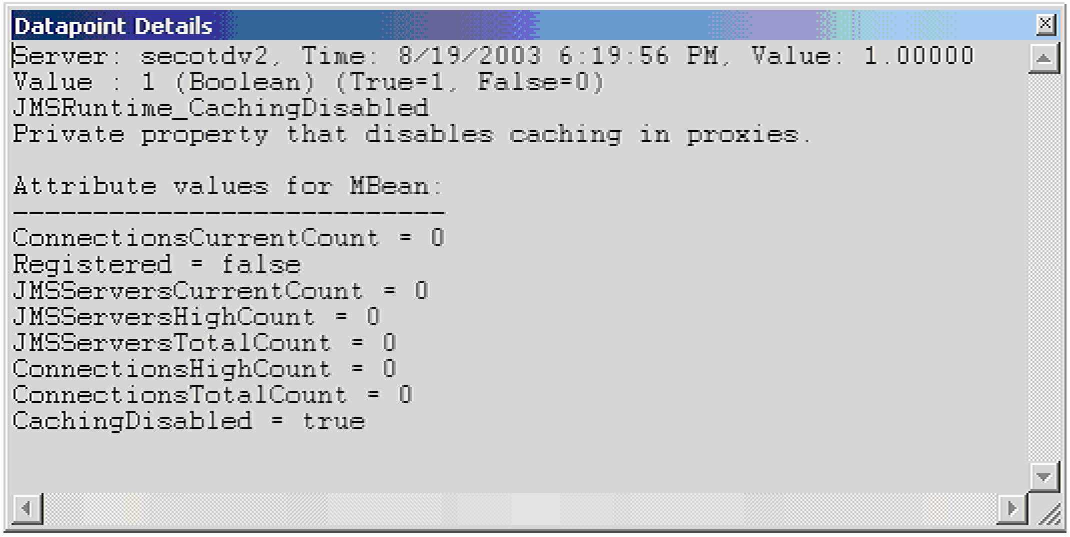
INTERVAL The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

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 6.x	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	No	Yes	Yes
BytesHighCount	Returns the peak number of bytes stored in the JMS server since the last reset.	Long	No	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	No	Yes	Yes
BytesReceivedCount	Returns the number of bytes received on this JMS server since the last reset.	Long	No	Yes	Yes
BytesThresholdTime	Returns the amount of time in the threshold condition since the last reset.	Long	No	Yes	Yes
CachingDisabled	Private property that disables caching in proxies.	Boolean	No	Yes	Yes
DestinationsCurrentCount	Returns the current number of destinations for this JMS server.	Long	No	Yes	Yes
DestinationsHighCount	Returns the peak number of destinations on this JMS server since the last reset.	Long	No	Yes	Yes
DestinationsTotalCount	Returns the number of destinations instantiated on this JMS server since the last reset.	Long	No	Yes	Yes
MessagesCurrentCount	Returns the current number of messages stored on this JMS server. This does not include the pending messages.	Long	No	Yes	Yes
MessagesHighCount	Returns the peak number of messages stored in the JMS server since the last reset.	Long	No	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	No	Yes	Yes
MessagesReceivedCount	Returns the number of messages received on this destination since the last reset.	Long	No	Yes	Yes
MessagesThresholdTime	Returns the amount of time in the threshold condition since the last reset.	Long	No	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	No	Yes	Yes

SessionPoolsCurrentCount	Returns the current number of session pools instantiated on this JMS server.	Long	No	Yes	Yes
SessionPoolsHighCount	Returns the peak number of session pools instantiated on this JMS server since the last reset.	Long	No	Yes	Yes
SessionPoolsTotalCount	Returns the number of session pools instantiated on this JMS server since the last reset.	Long	No	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:

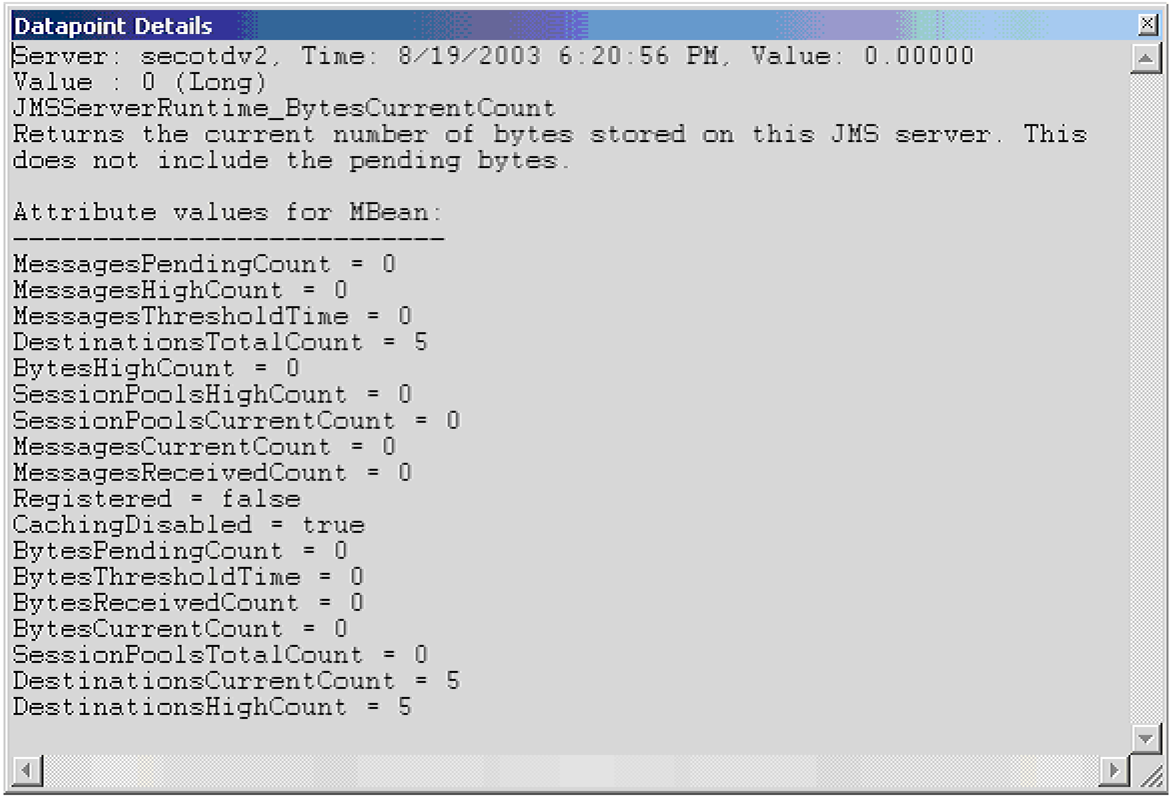
ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

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 6.x	WL 7.x	WL 8.x
BytesPendingCount	Returns the number of bytes pending (uncommitted and unacknowledged) for this session.	Long	No	Yes	Yes
BytesReceivedCount	Returns the number of bytes received by this session since the last reset.	Long	No	Yes	Yes
BytesSentCount	Returns the number of bytes sent by this session since the last reset.	Long	No	Yes	Yes

CachingDisabled	Private property that disables caching in proxies.	Boolean	No	Yes	Yes
ConsumersCurrentCount	Returns the current number of consumers for this session.	Long	No	Yes	Yes
ConsumersHighCount	Returns the peak number of consumers for this session since the last reset.	Long	No	Yes	Yes
ConsumersTotalCount	Returns the number of consumers instantiated by this session since the last reset.	Long	No	Yes	Yes
MessagesPendingCount	Returns the number of messages pending (uncommitted and unacknowledged) for this session.	Long	No	Yes	Yes
MessagesReceivedCount	Returns the number of messages sent by this session since the last reset.	Long	No	Yes	Yes
MessagesSentCount	Returns the number of bytes sent by this session since the last reset.	Long	No	Yes	Yes
ProducersCurrentCount	Returns the current number of producers for this session.	Long	No	Yes	Yes
ProducersHighCount	Returns the peak number of producers for this session since the last reset.	Long	No	Yes	Yes
ProducersTotalCount	Returns the number of producers for this session since the last reset.	Long	No	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	No	Yes	Yes
Transacted	Returns whether the session is transacted.	Boolean	No	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:

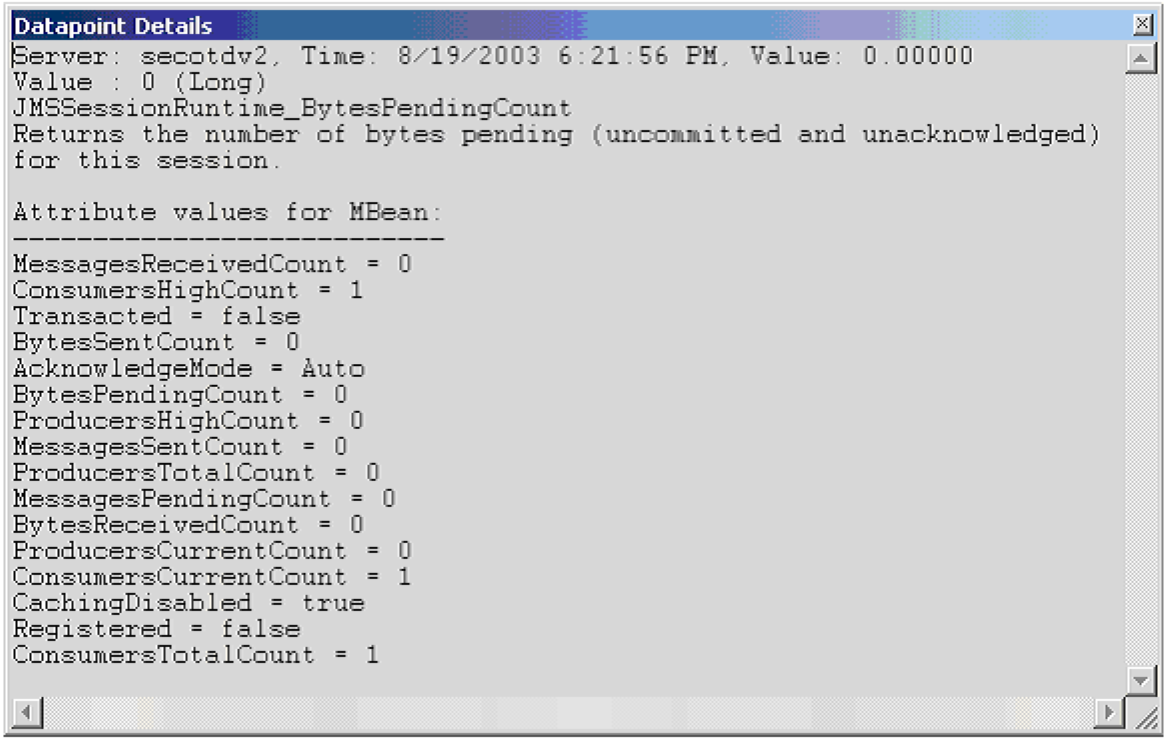
ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

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 6.x	WL 7.x	WL 8.x
Active	Returns whether the Transaction Recovery Service is currently activated on this server.	Boolean	No	Yes	Yes
CachingDisabled	Private property that disables caching in proxies.	Boolean	No	Yes	Yes
InitialRecoveredTransactionTotalCount	Returns the total number of transactions that are recovered from the Transaction Log initially.	Integer	No	Yes	Yes
RecoveredTransactionCompletionPercent	Returns the percentage of the initially recovered transactions that are completed.	Integer	No	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	No	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.

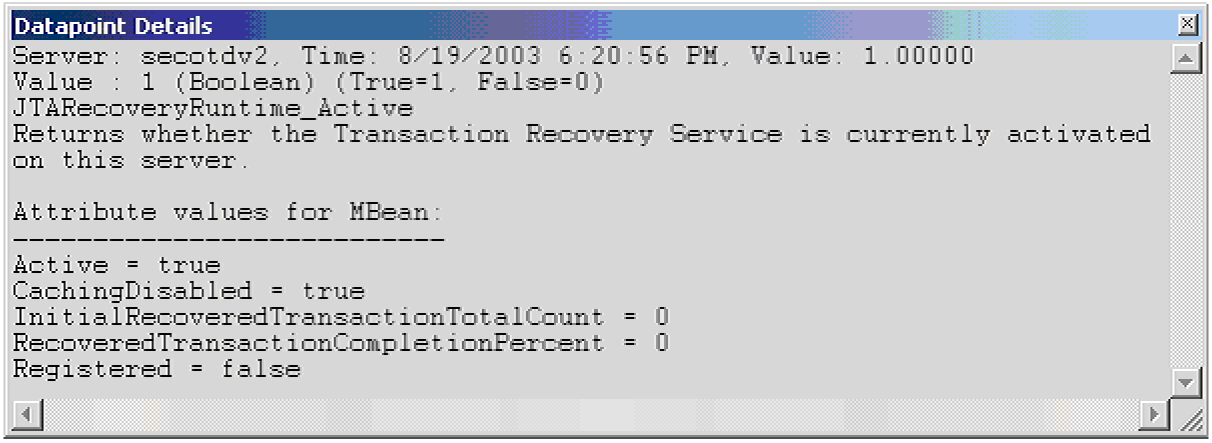
Data Point

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 6.x	WL 7.x	WL 8.x

ActiveTransactionsTotalCount	Returns the number of active transactions on the server.	Integer	Yes	Yes	Yes
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes	Yes
SecondsActiveTotalCount	Returns the total number of seconds for all committed transactions	Long	Yes	Yes	Yes
TransactionAbandonedTotalCount	Returns the number of transactions that were abandoned.	Long	Yes	Yes	Yes
TransactionCommittedTotalCount	Returns the number of committed transactions.	Long	Yes	Yes	Yes
TransactionHeuristicsTotalCount	Returns the number of transactions that completed with a heuristic status.	Long	Yes	Yes	Yes
TransactionRolledBackAppTotalCount	Returns the number of transactions that were rolled back due to an application error.	Long	Yes	Yes	Yes
TransactionRolledBackResourceTotalCount	Returns the number of transactions that were rolled back due to a resource error.	Long	Yes	Yes	Yes
TransactionRolledBackSystemTotalCount	Returns the number of transactions that were rolled back due to an internal system error.	Long	Yes	Yes	Yes
TransactionRolledBackTimeoutTotalCount	Returns the number of transactions that were rolled back due to a timeout expiration.	Long	Yes	Yes	Yes
TransactionRolledBackTotalCount	Returns the number of transactions that were rolled back.	Long	Yes	Yes	Yes

TransactionTotalCount	Returns the total number of transactions processed. This total includes all committed, rolled back and heuristic transaction completions.	Long	Yes	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.

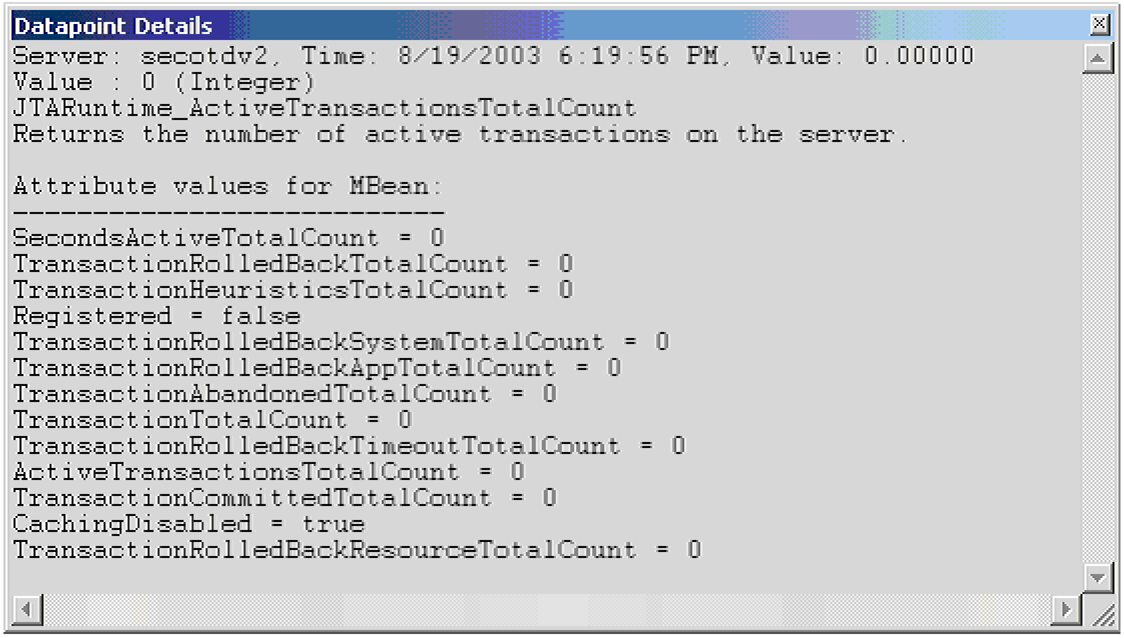
Data Point

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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	Yes
HeapFreeCurrent	Returns the current amount of free memory (in bytes) in the JVM heap.	Long	Yes	Yes	Yes
HeapSizeCurrent	Returns the current size (in bytes) of the JVM heap.	Long	Yes	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	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

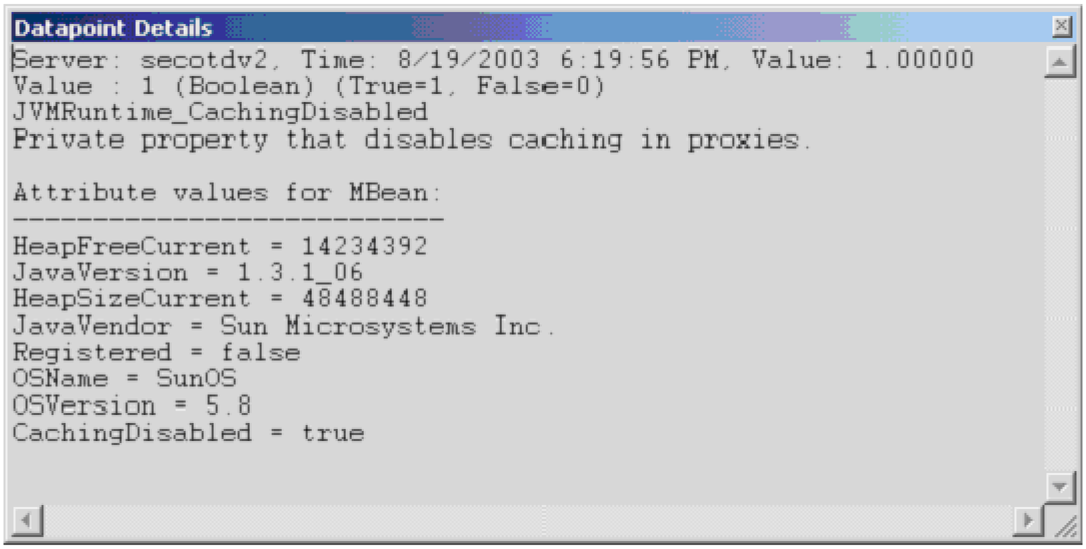
WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	Yes
MessagesLogged	Returns the total number of log messages generated by this instance of the WebLogic server.	Long	Yes	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	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:

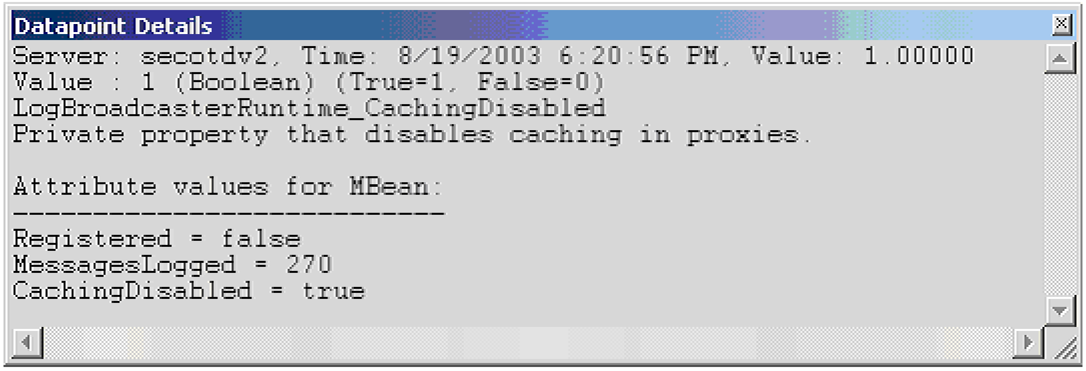
ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	No	Yes	Yes
JMSConnectionAlive	Returns the state of the EJB's JMS connection.	Boolean	No	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered	Boolean	No	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.

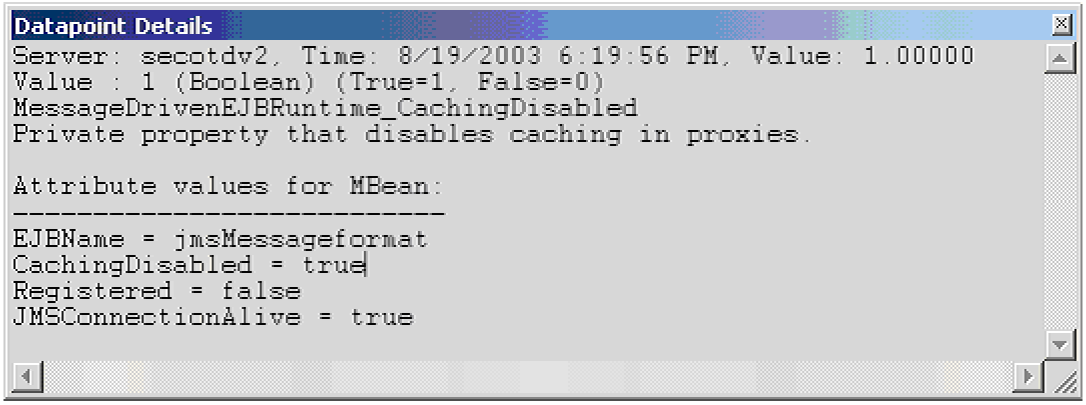
Data Point

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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	No	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	No	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

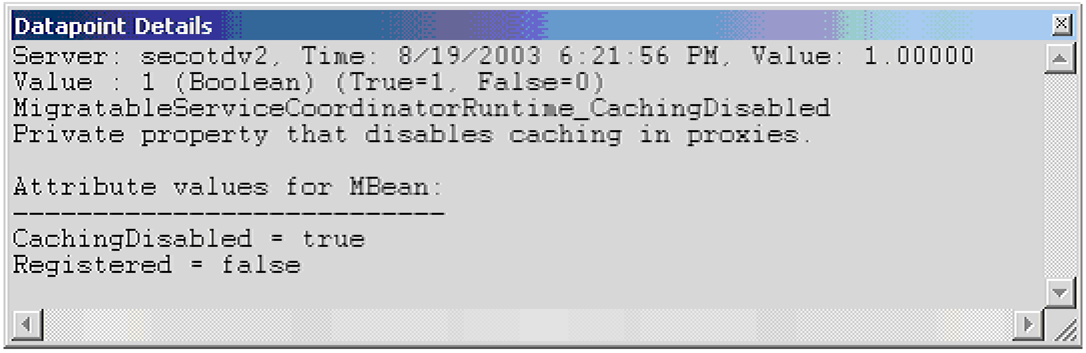
The name of the migratable service coordinator. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	No	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	No	Yes	Yes
StateVal	Returns an integer that identifies the current state of the server. Values range from 0 to 8.	Integer	No	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

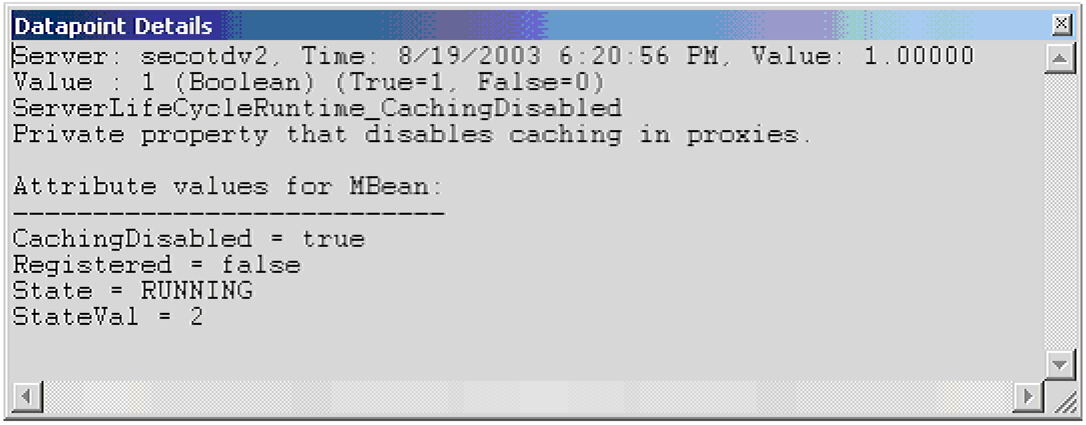
The application server name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

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 6.x	WL 7.x	WL 8.x
ActivationTime	Returns the time when the server was started.	Long	Yes	Yes	Yes

AdministrationPort	Returns the administration port on which this server is listening for connections.	Integer	No	Yes	Yes
AdministrationPortEnabled	Returns whether the AdministrationPort is enabled on the server.	Boolean	No	Yes	Yes
AdminServer	Checks if the server is an administrator server.	Boolean	No	Yes	Yes
AdminServerListenPort	Returns the port on which admin server is listening for connections.	Integer	Yes	Yes	Yes
AdminServerListenPortSecure	Returns the secureType on which admin server is listening for connections.	Boolean	No	Yes	Yes
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	Yes
ListenPort	Returns the port on which this server is listening for connections.	Integer	Yes	Yes	Yes
ListenPortEnabled	Returns whether the default ListenPort is enabled on the server.	Boolean	No	Yes	Yes
OAMVersion	Returns the OAM version info. Indicates release level of this server.	Integer	No	Yes	Yes
OpenSocketsCurrentCount	Returns the current number sockets registered for socket muxing on this server.	Integer	Yes	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes	Yes
RestartsTotalCount	Returns the total number of restarts for this server since the cluster was last activated.	Integer	Yes	Yes	Yes
SocketsOpenedTotalCount	Returns the total number of registrations for socket muxing on this server.	Long	Yes	Yes	Yes
SSLListenPort	Returns the port on which this server is listening for SSL connections	Integer	No	Yes	Yes
SSLListenPortEnabled	Returns if the default ISSListenPort is enabled on the server.	Boolean	No	Yes	Yes

StateVal	Returns current state of the server.	Integer	No	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:

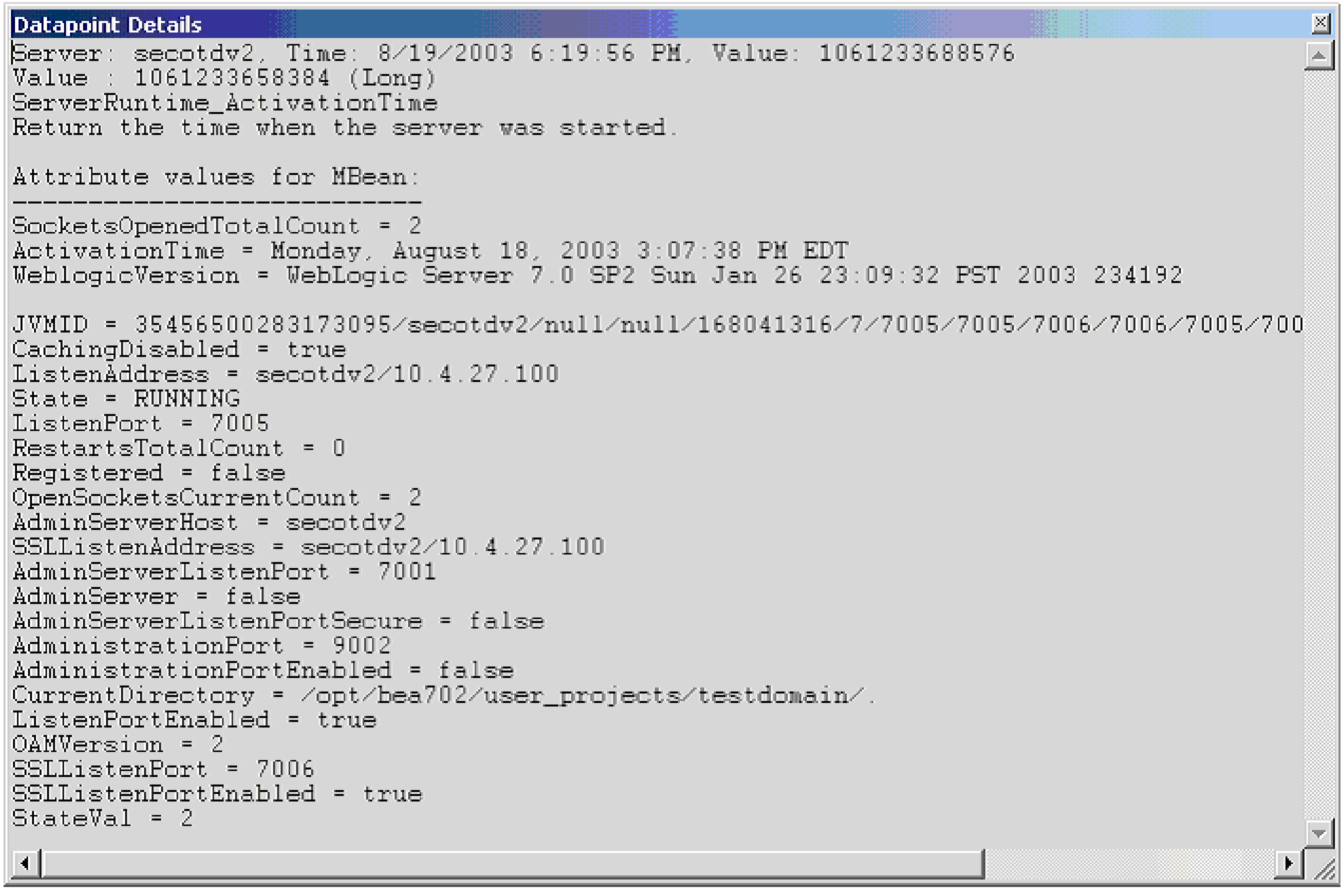
ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

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.



```
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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	Yes
InvalidLoginAttemptsTotalCount	Returns the cumulative number of invalid logins attempted on this server.	Long	Yes	Yes	Yes
InvalidLoginUsersHighCount	Returns the highest number of users with outstanding invalid login attempts for	Long	Yes	Yes	Yes

	this server.				
LockedUsersCurrentCount	Returns the number of currently locked users on this server.	Long	Yes	Yes	Yes
LoginAttemptsWhileLockedTotalCount	Returns the cumulative number of invalid logins attempted on this server while the user was locked.	Long	Yes	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes	Yes
UnlockedUsersTotalCount	Returns the number of times a user was unlocked on this server.	Long	Yes	Yes	Yes
UserLockoutTotalCount	Returns the cumulative number of user lockouts done on this server.	Long	Yes	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.

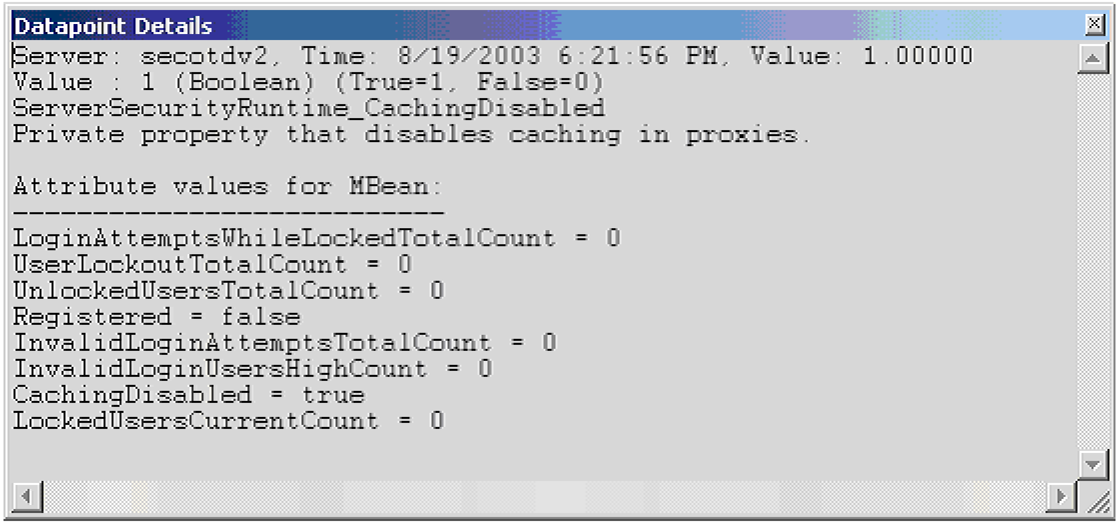
INTERVAL The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	Yes
ExecutionTimeAverage	Returns the average amount of time all invocations of the servlet have executed since it was created.	Integer	Yes	Yes	Yes
ExecutionTimeHigh	Returns the amount of time the single longest invocation of the servlet has executed since it was created.	Integer	Yes	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	Yes
ExecutionTimeTotal	Returns the amount of time all invocations of the servlet has executed since it was created.	Integer	Yes	Yes	Yes
InternalServlet	Returns whether this is an Internal Servlet.	Boolean	No	No	Yes
InvocationTotalCount	Returns the total number of times the servlet has been invoked.	Integer	Yes	Yes	Yes
PoolMaxCapacity	Returns the maximum capacity of this servlet for single thread model servlets.	Integer	Yes	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered	Boolean	Yes	Yes	Yes
ReloadTotalCount	Returns the total number of times the servlet has been reloaded.	Integer	Yes	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:

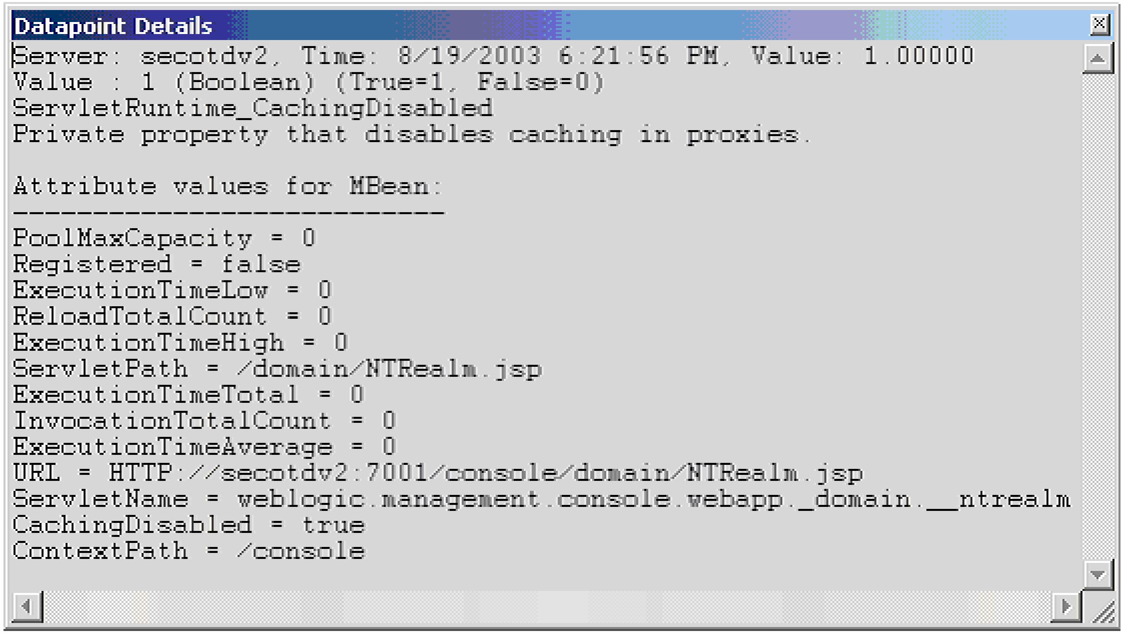
ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

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.



```
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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	No
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	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

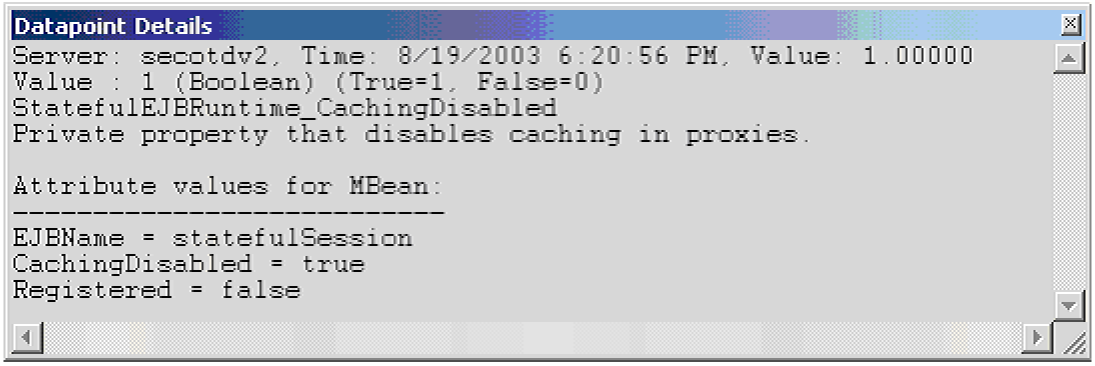
The remainder of the EJB name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	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

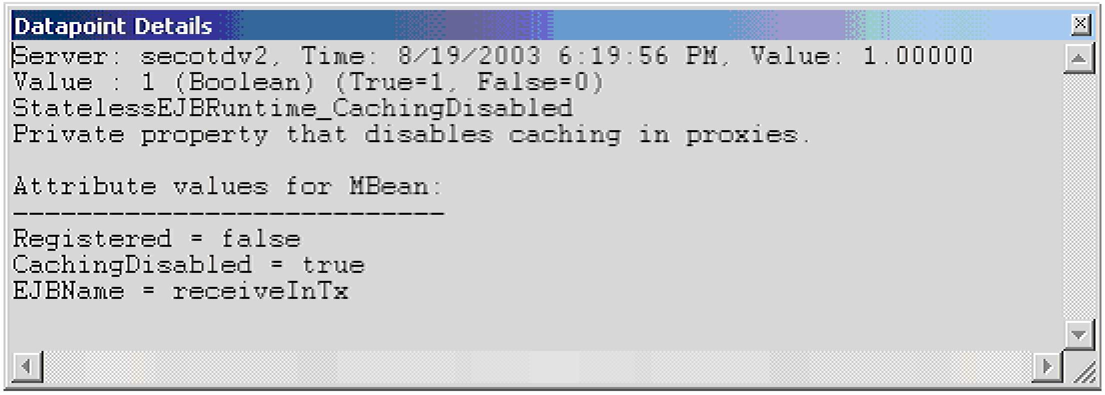
The remainder of the EJB name. You can specify one or more names for monitoring. In any combination, select values from the discovered list, or enter values manually.

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	WL	WL

			6.x	7.x	8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	Yes
ExceptionCount	Returns the total number of exceptions thrown while executing scheduled triggers.	Integer	Yes	Yes	Yes
ExecutionCount	Returns the total number of triggers executed.	Integer	Yes	Yes	Yes
ExecutionsPerMinute	Returns the average number of triggers executed per minute.	Integer	Yes	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes	Yes
ScheduledTriggerCount	Returns the number of currently active scheduled triggers.	Integer	Yes	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

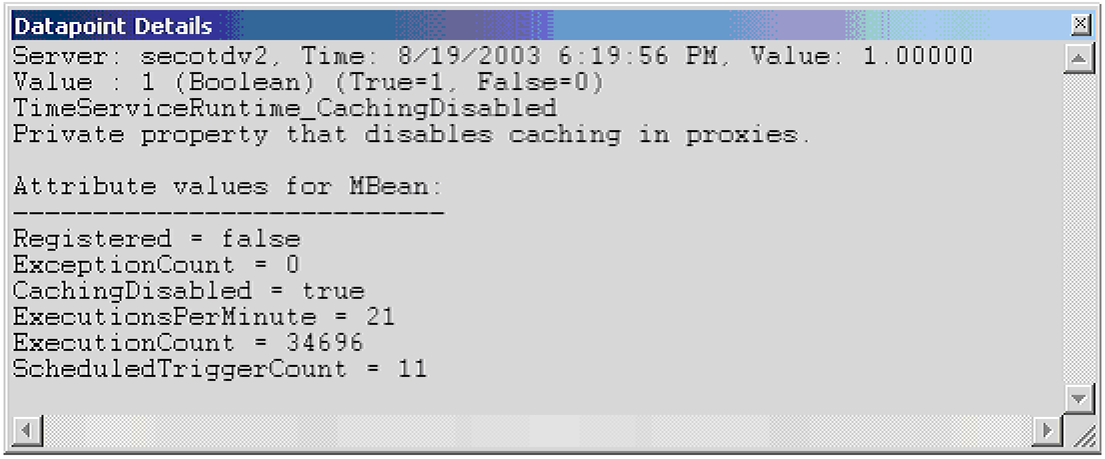
task interval and the raw data value of the counter in the current task interval.

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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	No	Yes	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	No	Yes	Yes
TransactionCommittedTotalCount	Returns the number of committed transactions.	Long	No	Yes	Yes

TransactionHeuristicCommitTotalCount	Returns the number of transactions for which this resource has returned a heuristic commit decision.	Long	No	Yes	Yes
TransactionHeuristicHazardTotalCount	Returns the number of transactions for which this resource has reported a heuristic hazard decision.	Long	No	Yes	Yes
TransactionHeuristicMixedTotalCount	Returns the number of transactions for which this resource has reported a heuristic mixed decision.	Long	No	Yes	Yes
TransactionHeuristicRollbackTotalCount	Returns the number of transactions for which this resource has returned a heuristic rollback decision.	Long	No	Yes	Yes
TransactionHeuristicsTotalCount	Returns the number of transactions that completed with a heuristic status.	Long	No	Yes	Yes
TransactionRolledBackTotalCount	Returns the number of transactions that were rolled back.	Long	No	Yes	Yes
TransactionTotalCount	Returns the total number of transactions processed. This total includes all committed, rolled back and heuristic transaction completions.	Long	No	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:

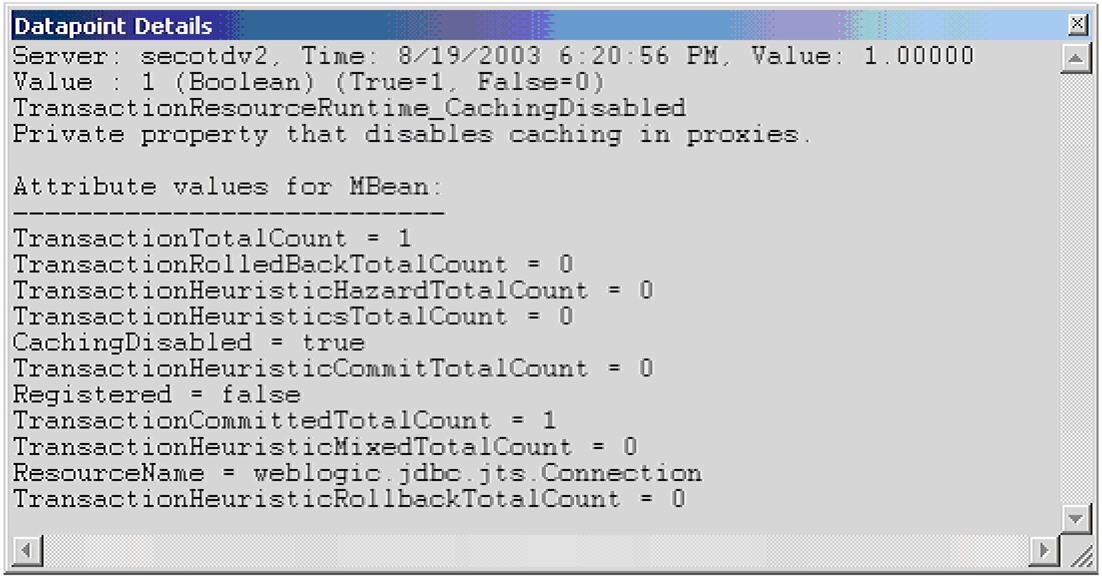
ACTUAL	The counter returns the raw data value.
INTERVAL	The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

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 6.x	WL 7.x	WL 8.x
CachingDisabled	Private property that disables caching in proxies.	Boolean	Yes	Yes	Yes
DeploymentState	Returns the current deployment state of the module.	Integer	No	No	Yes
IndexDirectoryDisabled	Returns the directory indexing indicator configured in weblogic.xml.	Boolean	No	No	Yes
JSPDebug	Returns the JSP's debug/line numbers parameter values configured in weblogic.xml.	Boolean	No	No	Yes
JSPKeepGenerated	Returns the JSP's KeepGenerated parameter value configured in weblogic.xml.	Boolean	No	No	Yes
JSPPageCheckSecs	Returns the JSP's PageCheckSecs value configured in weblogic.xml.	Long	No	No	Yes
JSPVerbose	Returns the JSP's Verbose parameter value configured in weblogic.xml.	Boolean	No	No	Yes
OpenSessionsCurrentCount	Returns the current total number of open sessions in this component.	Integer	Yes	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	Yes

Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	Yes	Yes	Yes
ServletReloadCheckStatus	Returns the servlet reload check seconds configured in weblogic.xml.	Integer	No	No	Yes
SessionCookieMacAgentSecs	Returns the session's cookie max age configured for http sessions.	Integer	No	No	Yes
SessionInvalidationIntervalSecs	Returns the invalidation check timer interval configured for http sessions.	Integer	No	No	Yes
SessionMonitoringEnabled	Returns the session monitoring indicator configured in weblogic.xml.	Boolean	No	No	Yes
SessionsOpenedTotalCount	Returns the total number of sessions opened in this server.	Integer	Yes	Yes	Yes
SessionTimeoutSecs	Returns the timeout configured for http sessions.	Integer	No	No	Yes
SingleThreadServletPoolSize	Returns the single threaded servlet pool size configured in weblogic.xml.	Integer	No	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.

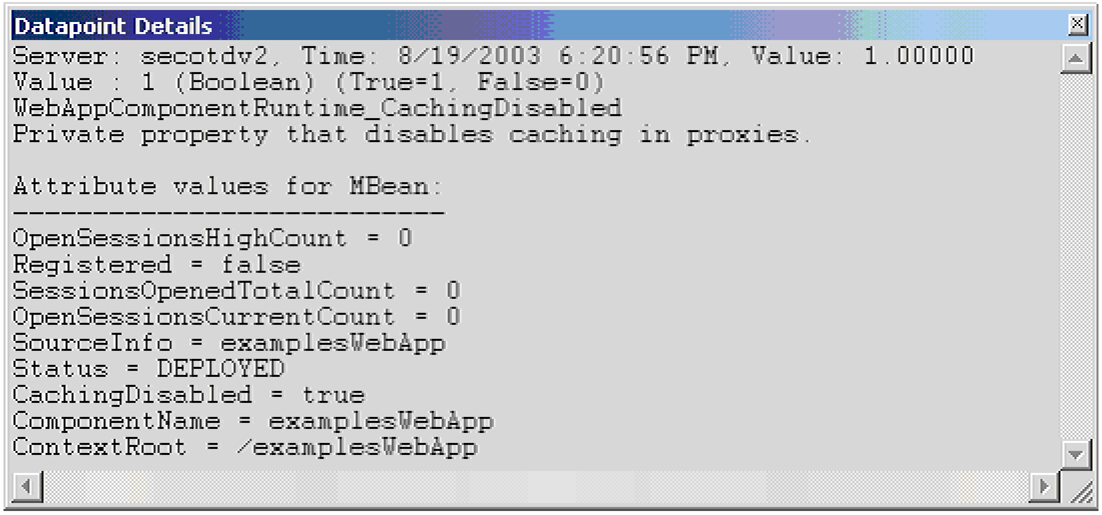
INTERVAL The counter returns the difference between the raw value of the counter in the last task interval and the raw data value of the counter in the current task interval.

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 6.x	WL 7.x	WL 8.x

CachingDisabled	Private property that disables caching in proxies.	Boolean	No	Yes	Yes
DefaultWebServer	Returns whether it is the defaultWebServer or a VirtualHost.	Boolean	No	No	Yes
Registered	Returns false if the MBean represented by this object has been unregistered.	Boolean	No	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

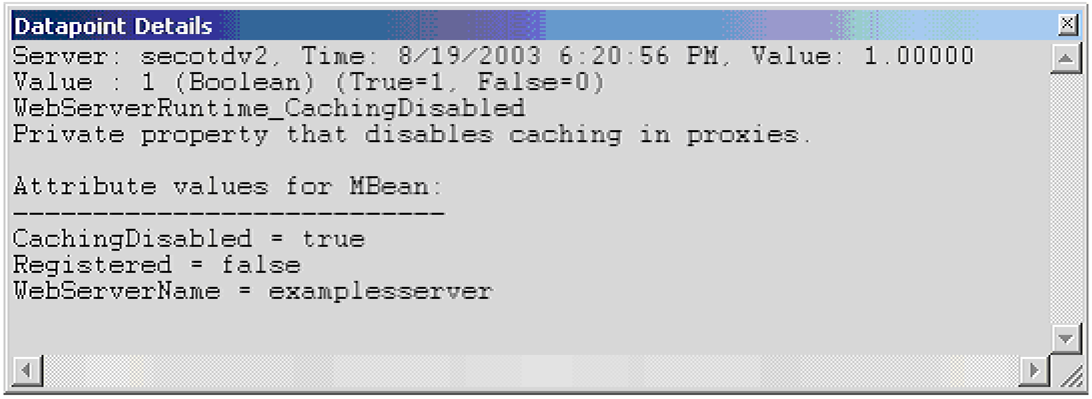
WebLogic Application Server where the instance you want to monitor resides. You can specify one or more servers for monitoring. In any combination, select values from the discovered list, or enter values manually.

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.

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 Bean Module
- WebSphere Cache Module
- WebSphere Connection Pool Module
- WebSphere J2C Module
- WebSphere JVM Runtime Module
- WebSphere ORB Perf Module
- WebSphere Scheduler Module
- WebSphere Servlet Sessions Module
- WebSphere System Module
- WebSphere Thread Pool Module
- WebSphere Transaction Module
- WebSphere Web App Module
- 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 driven beans).	5.0	Low	Long
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	5.0 and above	Low	Long

		entries.			
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 and/or disk because their timeout has expired.	5.0 and above	Low	Long
InMemoryAndDiskCacheEntryCount	Entries	Current number of cache entries created from	5.0 and above	Low	Long

		this template. Refers to the per-template equivalent of totalCacheSize.			
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 invalidations received for	5.0 and above	Low	Long

		this template where no corresponding entry exists.			
LocalExplicitInvalidationCount	explicitInvalidationsLocal	Explicit invalidations generated locally, either programmatically or by a cache policy.	5.0 and above	Low	Long
RemoteExplicitInvalidationCount	explicitInvalidationsRemote	Explicit invalidations received from a cooperating JVM in this cluster.	5.0 and above	Low	Long
RemoteCreationCount	remoteCreations	Entries received from cooperating dynamic caches.	5.0 and above	Low	Long

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.

Information presented in the table on this page is:
Reprinted Courtesy of International Business Machines Corporation copyright (2005) (c) International Business Machines Corporation.

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	5.0	Medium	Long

		the JDBC driver.			
	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 other than the deployment manager.	6.0 and above	High	Stat
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.

Information presented in the table on this page is:
Reprinted Courtesy of International Business Machines Corporation copyright (2005) (c) International Business Machines Corporation.

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	Number of garbage collection calls. This counter is not available unless -	4.0 and above	Max	Long

	calls	XrunpmiJvmpiProfiler is set when starting the JVM.			
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.

Information presented in the table on this page is:
Reprinted Courtesy of International Business Machines Corporation copyright (2005) (c) International Business Machines Corporation.

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 maximum session count.	5.0	Low	Long
CacheDiscardCount	cacheDiscards	Number of session objects that have	5.0	Low	Long

		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.			
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	Medium	Long
ExternalReadSize	externalReadSize	Size of session data read from persistent store. Applicable only for (serialized) persistent sessions; similar to externalReadTime above.	5.0	Medium	Long
ExternalWriteTime	externalWriteTime	Time (milliseconds) taken to write the session data from the persistent store. Applicable only for (serialized) persistent sessions. Similar to externalReadTime described above.	5.0	Medium	Long
ExternalWriteSize	externalWriteSize	Size of session data written to persistent	5.0	Medium	Long

		store. Applicable only for (serialized) persistent sessions. Similar to externalReadTime described above.			
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	timeSinceLastActivated	The time difference in milliseconds between previous and current access time stamps. Does not include session time out.	5.0	Medium	Long
TimeoutInvalidationCount	invalidatedViaTimeout	The number of requests for a session that no CountStatistic exists, presumably because the session timed out.	5.0	Low	Long
ActivateNonExistSessionCount	attemptToActivateNot ExistentSession	Number of requests for a session that no longer exists, presumably because	5.0	Low	Long

		the session timed out. Use this counter to help determine if the timeout is too short.			

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 many operating	5.0	Low	Long

		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.

Information presented in the table on this page is:
Reprinted Courtesy of International Business Machines Corporation copyright (2005) (c) International Business Machines Corporation.

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 transactions.	4.0 and above	Medium	Stat
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.

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 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 request or reply.	5.02 and above	Medium	Stat
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

Channel Status

Error Log Entries

Percent Queue Depth

Performance Events

Queue Depth

Queue Manager Connections

Queue Manager Events

Queue Manager Statistics

Queue Manager Up/Down

Queue Statistics

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	The BaseQName in the alias queue attributes is not recognized as a

Queue	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 =	Controls whether or not trigger messages are written to an initiation

<integer>	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 /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.

.

About Monitoring Templates

Using 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 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:

- SAP
- Server Analysis Agent
- SNMP
- WebLogic
- WebSphere
- WebSphere MQ
- Windows Registry
- WMI

Note: You cannot modify pre-defined 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

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

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 is granted.
	connectionPoolModule.concurrentWaiters	WebSphere extended counter for monitoring connectionPoolModule.concurrentWaiters
	connectionPoolModule.faults	Average waiting time in milliseconds until a connection is granted.

	connectionPoolModule.percentMaxed	Average percent of the time that all connections are in use. Rule: IF 'WebSphere connectionPoolModule.connectionPoolModule.percentMaxed(Node: "**", 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

QALoad provides the following pre-defined WMI templates:

QALoad-Active Monitoring Availability

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	Current Blocked Async I/O Requests	

Services Global		
	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 Sessions Errored Out Sessions Forced Off Sessions Logged Off Sessions Timed Out	The Microsoft Windows Availability template uses these 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 Counters

Adding Counters to a Task Using New Discovery Data

You can add counters to a monitoring task by generating the counter data available for the task and selecting the counters and instances to add to the task.

To counters and instances to a monitoring task:

1. Click **Tools>Monitoring>Add counter>Use new discovery data**. The Edit Monitoring Machine Wizard appears.
2. Follow the instructions for using the wizard to discover and add counters to your monitoring task.

Adding Counters to a Task Using Cached Discovery Data

You can add counters to a monitoring task using cached discovery data.

Select the counter to add or modify:

1. Select **Monitoring>Add counter>Use cached discovery data**. This Add/Edit Counters dialog box appears.
2. From the **Available Items** pane, select the Template tab or the Counter tab.
3. Select a template or a counter to monitor in the task for this machine and monitor type, and click **Add**. The items you select display in the **Selected Items** pane. Click **Add All** to add all the items on the selected tab to the Selected Items pane.
4. To remove an item, select the item in the Selected Items pane and click Remove. The items is returned to the Available Items pane.

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..

5. Repeat this process until you select all the templates and counters you want to monitor in this task.
6. Click **Next**. The Choose Instances dialog box displays.

Choose the instances of the counter to monitor:

When you click Next in the previous dialog box, the Choose Instances dialog box appears.

1. Review the counters you selected. When a red dot appears next to a counter, you must 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 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.

Save the task:

When you click Next in the previous dialog box, the Summary dialog box appears.

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 **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. On the **Monitoring Options** tab, select the monitor, counter, or counter family to delete.
2. Click **Tools>Monitoring>>Remove monitor/Remove 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.

Managing Monitoring Templates

Creating a New Template

To access the New Monitoring Template wizard:

Click Tools>Monitoring>Manage monitoring templates>New monitoring template. The New Monitoring Template wizard appears. Click Next to start the procedure.

Use the following steps in the New Monitoring Template wizard to create a new monitoring template:

1. Enter the template properties
2. Configure the monitor
3. Counter discovery
4. Choose the counters
5. Choose the instances
6. Review, save, and create the template

Using an Existing Template

Use the following steps to apply a previously created or pre-defined template.

To apply an existing template:

1. Select **Tools>Monitoring>Manage Monitoring Templates>Use existing template**. The Select a Monitor Template File dialog box displays.

2. Double-click a template type, then select a template and click **Open**. The template is applied to the monitoring task.

Editing Instances for Templates

To edit instances in a template:

1. On the Monitoring Options tab, select the template to edit.
2. Click **Tools>Monitoring>Manage monitoring templates>Edit instances**. The Edit Template Instances Wizard appears.

To 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.

To 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 for Custom Templates

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.

Select the counter to add or modify:

1. Select **Monitoring>Manage monitoring templates>Add/Edit counter>Use cached discovery data**. The Edit Template Counters wizard appears with the Add/Edit/Remove Template Counters dialog box displayed.
2. From the **Available Items** pane, select the Template tab or the Counter tab.
3. Select a template or a counter to monitor for this machine and monitor type, and click **Add**. The items you select display in the **Selected Items** pane. Click **Add All** to add all the items on the selected tab to the Selected Items pane.
4. To remove an item, select the item in the Selected Items pane and click Remove. The items is returned to the Available Items pane.

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.

5. Repeat this process until you select all the templates and counters you want to monitor.
6. Click **Next**. The Choose Instances dialog box displays.

Choose the instances of the counter to monitor:

1. Review the counters you selected. When a red dot appears next to a counter, you must 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**. You return to the **Choose Instances** dialog box.
7. Repeat this process for each designated counter.
8. Click **Next**. The Summary dialog box displays.

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 **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. On the **Monitoring Options** tab, select the counter or counter family to delete.
2. Click **Tools>Monitoring>Manage monitoring templates>Remove counter**.
3. When the verification dialog box displays, click **OK**.

Note: You cannot remove the last counter in a family.

Creating and Editing Monitoring Tasks

Creating a New Monitoring Task

To access the New Monitoring Task wizard:

Click Tools>Monitoring>New monitoring task. The New Monitoring Task Wizard appears. Click Next to start the procedure.

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 counters

4. Choose the counters for the monitoring task
5. Choose the instances of the counter to monitor
6. Review the monitor definition
7. Save and create the monitoring task

Using an Existing Monitoring Task

To select an existing monitoring task:

1. Click **Tools>Monitoring>Use existing task**. The Choose an Existing Task dialog box appears.
2. Select a task and click **OK**. The task displays in Conductor's Monitoring Options tab.

Adding a Monitoring Machine

To access the New Monitoring Task wizard:

Click Tools>Monitoring>Add monitor. The Add Monitoring Machine wizard appears. Click Next to start the procedure.

Use the following steps in the Add Monitoring Machine wizard to add a monitoring machine to the task:

1. Enter properties of the monitoring machine
2. Configure the monitor
3. Discover the counters
4. Choose the counters for the monitoring task
5. Choose the instances of the counter to monitor
6. Review the monitor definition
7. 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 access the Edit Monitoring Machine wizard:

Click Tools>Monitoring>Edit monitor. The Edit Monitoring Machine wizard appears. Click Next to start the procedure.

Use the following steps in the Edit Monitoring Machine 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 Instances

Use the following procedure to edit instances if the counters you are monitoring.

To access the Edit Instances dialog boxes:

1. On the Monitoring Options tab, select the machine, the counter, or the instance to edit.
2. Click **Tools>Monitoring>Edit instances**. The Edit Instances dialog boxes display.

To 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.

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.

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.

Server Analysis Agent

Server Analysis Agents

Server Analysis agents use enhanced ServerVantage technology to provide server utilization data without a complete ServerVantage deployment. Server Analysis agents, provided on the QACenter Performance Edition CD, are quickly and easily installed on the servers that you wish to monitor during a load test. Server Analysis agents provide you with valuable server utilization metrics — called counters — on Web servers, application servers, and database servers being exercised by your load test to help you to pinpoint performance bottlenecks when load testing.

Unlike a full ServerVantage installation, you can start, stop, and configure Server Analysis Agents right from the familiar interface of the QALoad Conductor.

Server utilization data from the agents and response time information from QALoad is all automatically downloaded and correlated through the use of ActiveAnalysis, and is available for post-test analysis through QALoad Analyze.

Server Analysis agents are set up when you create a monitoring task. For details about using Server Analysis agents in a load test, see Creating a New Monitoring Task.

Server Analysis Agent Templates

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	
Memory	Available Bytes	
	Cache Faults/sec	
	Committed Bytes	
	Page Faults/sec	
Paging File	% Usage Peak	

Processor	% Processor Time	
	% User Time	
	Interrupts/sec	
Server	Bytes Total/sec	

ServerVantage

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. For more information about installing or configuring ServerVantage refer to its product documentation.
- ServerVantage must be scheduled to monitor the specified performance counters at a time that coincides with a running *QA*Load test
- *QA*Load must be able to access the appropriate Agent stations to collect resource utilization data at the end of the load test.

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.

This integration uses standard Netbios ports. Netbios ports 137-139 and 145 should be kept open when performing this type of activity. 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

To set up integration with ServerVantage:

1. On the Conductor Test Information Screen, click the **Monitoring Options** tab.
2. Click the **Set up monitoring** link, then select **Set up a new monitoring task**.
3. In the **Define Monitor** dialog box, click the arrow in the **Monitor Type** box and select **ServerVantage**.
4. In the **Control Server Database Host** field, type the hostname of the Database server machine.
5. Click **Next**. The **Configure Monitor** dialog box displays.
6. In the **Username** field, type a valid Windows admin login user name for the Database server machine.
7. In the **Password** field, type the password that corresponds to the user name above, if necessary.

8. Select the **Override Default Database** check box to provide the ServerVantage database name. When this option is not selected, *QA*Load 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.
9. In the Vantage Agent Configuration area, type the ServerVantage Agent name, and click **Add** to add it to your load test.
10. Click **Next** to proceed to the next step.

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 *QA*Load Conductor. You can install either the ApplicationVantage Agent or the ApplicationVantage Remote Agent.

! You have specified on which NIC to capture from the Machine Assignment tab 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 9.5 or greater. You must select the ApplicationVantage option, and then set the Network Interface Card (NIC) ServiceName 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) ServiceName.

To enable ApplicationVantage:

1. Click Conductor's **Screen Assignment** tab.
2. Click in the **Script** column to enable the **Browse (...)** button. Then, click **Browse**. The **Select Script** dialog box displays.
3. Click the **ApplicationVantage Mode** box at the bottom of the dialog box..
4. Click **OK**.

5. Set the NIC ServiceName.

Setting up Network Interface Card ServiceName

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 ServiceName:

1. With the session ID you want to use for your test open, click on the Conductor's **Machine Assignment** tab.
2. Click the **Manage Player Machines** button for the Conductor to query your test network for installed Player Agents. You also can select **Tools>Manage Player Machines**. The **Manage Player Machines and Groups** dialog box displays with names of available Player machines.
3. In the **All Player Machines and Groups** area, double-click the Player machine that will be running the virtual user to be captured. The **Edit Player Machine** dialog box displays.
4. Click the **Application Vantage Settings** tab.
5. From the drop-down list in the **NIC ServiceName** field, select the NIC that is used by the machine.
6. Click **Apply**, then click **OK.**

Troubleshooting

Conductor pre-test checks

Before a test begins, the Conductor completes the following pre-test checks of the parameter files and Players. If any of these checks fail, the Conductor displays an error message.

- Are there enough Players configured to support the number of users specified in the session ID file?
- Does the number of users specified in the session ID file exceed the maximum number of users defined by your authorization key?
- Can the specified compiled script files be accessed?
- Are all Players communicating with the Conductor? (The Conductor sends a request message to all the Players to verify that they are up and running.)

Executing SSL scripts that use client certificates

If you are executing SSL scripts that use client certificates, you must manually copy the client certificates in use to the Player machine(s) executing the script(s).

Manually copy the client certificates from the `\Program Files\Compuware\QALoad\Certificates` directory to the same default directory on the Player machine.

Note: On the Unix player platform, you must create the `Certificates` sub-directory in the `QALoad` directory. The directory name is case sensitive.

Heartbeat message failure on a virtual user

When a Player machine crashes or experiences a loss of communication, the heartbeat message that the Conductor sends out (if enabled) fails. This situation is indicated in the runtime Conductor through a message on each virtual user that is affected. When the heartbeat message fails for a virtual user, the Status column of the Details view of a script displays the following message: "The Player running this user failed to respond to a heartbeat message."

The option for enabling a heartbeat message is located on the Player tab of the Options dialog box in the Conductor.

Timing file is too big

Depending on the length of the load test and the amount of data that was collected, timing files can grow to excessively large sizes that become difficult to handle. To prevent timing files from becoming too large, try modifying the following settings:

- Disable automatic middleware checkpoint timings in the Conductor
- Use the Conductor's timing data thinning options

Both of these settings are located on the Timing Options dialog box, which can be accessed from the Script Assignment tab of the Conductor.

Tips for running QALoad tests on UNIX systems

To successfully run large QALoad tests on UNIX systems, you may need to make adjustments to your settings as described below:

General (AIX, Solaris, and RedHat Linux)

When you attempt to run a large number of virtual users on UNIX platforms, the virtual users do not always synch. If virtual users do not synch, try increasing the Virtual User Startup Delay. By default, QALoad Conductor sets the VU Startup Delay to 1 millisecond. This default is not high enough for UNIX platforms. If the UNIX Player receives a value less than 15 milliseconds, the delay will be 15 milliseconds or more.

To increase the delay:

1. In the *QA*Load Conductor, click **Tools>Options**.
2. Click the **Player** tab.
3. In the **VU Startup Delay** field, type the number of milliseconds to delay virtual user startup.

Solaris

The default file descriptor limit on Solaris has a "soft" limit of 64, and a "hard" limit of 1024 (Solaris 2.6). Per the Solaris 2 FAQ (refer to http://www.wins.uva.nl/pub/solaris/solaris2.html), the file descriptor limit is described in the getrlimit() manual page as: "One more than the maximum value that the system may assign to a newly created descriptor. This limit constrains the number of file descriptors that a process may create."

To increase this limit, system administrators can modify the `/etc/system` file and reboot the system. For example:

```
* set hard limit on file descriptors

set rlim_fd_max - 4096
```

```
*set soft limit on file descriptors
set rlim_fd_cur = 1024
```

Index