Getting Started, Running a Load Test, and Analyzing Results

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

Technical Support Hotline:
1-800-538-7822

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

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++ or Java-based script. For more information, see Overview of the Script Development Workbench.

Conductor

Control session activity. Initiate and monitor the test. For more information, see Overview of the QALoad Conductor.

Player

Simulate user activity based on commands given from the Conductor. For more information, see Overview of the QALoad Player.

Analyze

View summary report data and create other statistical reports from the test. For more information, see Overview of QALoad Analyze.

Before you begin using QALoad, you should familiarize yourself with load testing concepts. For more information, see The Load Testing Process.

How to Use the Online Help

QALoad's online help includes a Table of Contents (TOC), an index, and a search input field. When you click on a topic title in the TOC, the topic appears in the right-hand pane of the online help. Topics included in the Table of Contents are synchronized with the TOC. This means that when you click a topic link in the right-hand pane, the topic title is highlighted in the TOC in the left-hand pane.

Use the Index tab to search for a specific reference by typing a keyword in the index entry field. Use the Search tab to search for associated topics. Type a keyword in the search entry field. All topics that contain the keyword are listed.

When you are working in QALoad, click the Help button on an individual dialog box or press F1 to view a description of each field in the dialog box. For these topics and others that do not appear in the Table of Contents, the synchronization function with the TOC is disabled.

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 in the Machines window 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. For more information, see Machines Dialog Box.

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

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, 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.

Validating Scripts

Before you conduct an actual load test, you should individually validate the script(s) you plan to use by running a simple load 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:

Validating a Script in the Script Development Workbench

Validating a Script in the Player

Validating a Script in the Conductor

Running a Load Test and Analyzing Results

Running a Load Test

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

To start a load test:

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

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

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

Running a Series of Tests

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

Monitoring a Load Test

When you start a test, 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. For more information, see Runtime Window Interface.

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
- ! Exit, abort, or quit the test

For more information, see Runtime Toolbar Buttons.

Stopping a Load Test

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

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

To stop a load test:

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

Adding Post-test Comments

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

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 on any machine with a Web browser.

When running a test using a particular session ID file (set up in the Conductor), each Player compiles a local timing file that comprises 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. These values specify 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

Use QALoad Analyze 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 manipulating any data. All the files necessary for these 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. This is determined by the QALoad Conductor option you select when you run the load test.

Adding Post-test Comments

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

To configure the Conductor for adding post-test comments:

1. Select **Tools>Options**. The Options dialog box appears.

2. In the **Dialog** section of the Conductor Sessions page , select **Display post test comments dialog**.

3. Click **OK**. The Conductor is now configured so that you can add comments when a test completes.

To add post-test comments:

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

   OR

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

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

3. *QA*Load Analyze.

Index