*QA*Load

Testing User's Guide

Release 05.01

COMPUWARE®

Please direct questions about *QA*Load
or comments on this document to:

QALoad Customer Support
Compuware Corporation
One Campus Martius
Detroit, MI 48226-5099

1-800-538-7822

Outside the USA and Canada, please contact
your local Compuware office or agent.

Doc. CWQLUX510
April 06, 2004

Table of Contents

Summary of Changes

For up-to-date details about new features, known issues, and recent updates in this release, please refer to the *QA*Load Release Notes. Access the Release Notes from the Windows **Start** menu by choosing **Programs\Compuware\*QA*Load\Release Notes**.

Introduction

*QA*Load is a high performance tool that allows for automated load testing of client/server-based systems and associated networks. A *QA*Load test eliminates the need and expense of having rooms of users testing new systems before going live. This guide includes the following chapters to assist you in performing client/server load and stress testing using *QA*Load:

- **Chapter 1: QALoad Testing Overview —** Provides an overview of the testing process and descriptions of the script development and testing components.
- **Chapter 2: Getting Started —** Describes how to prepare *QA*Load's Player and Conductor components to run a load test.
- **Chapter 3: Running a Load Test —** Describes how to prepare for, start, monitor, and stop a load test.
- **Chapter 4: Including QARun Scripts in Load Tests —** Describes how to create a test script that includes *QA*Run transactions.
- **Chapter 5: Load Testing the Network with NetLoad —** Describes how to use *QA*Load's NetLoad component to generate traffic on your network in a controlled manner and gather performance timings from the network.
- **Chapter 6: Analyzing Load Test Results —** Provides information for analyzing test results using *QA*Load's Analyze component.

Who Should Read This Guide

The *QALoad Testing User's Guide* is intended to be a general reference guide for *QA*Load users. It provides a basic product and component overview and explains how to configure the system, set up tests, run tests, and view the results of a test. It also explains how to incorporate *QA*Run scripts into load tests, and how to use *QA*Load's NetLoad component to load the network.

The *QALoad Testing User's Guide* does not include reference or procedural information to assist you in preparing test scripts. That information, including special techniques for inserting variable information into your scripts, is detailed in the *QALoad Script Development Guide*.

We recommend that you use the *QALoad Testing User's Guide* to familiarize yourself with the *QA*Load testing process and to set up your test. Then, use the *QALoad Script Development Guide* to create your test script(s). When you are finished creating test scripts, use the procedures detailed in the *QALoad Testing User's Guide* to run your test and analyze the results.

Related Publications

In addition to this guide, the *QA*Load documentation set includes the following related publications:

- *QALoad Script Development Guide* provides procedural information for creating test scripts for your application. It is designed to guide you through the preparation of a test script, including recording a transaction, converting it to a reusable test script, and modifying the code to accommodate variable information and other special circumstances.
- *QA*Load's online help facilities provide field-level and overview information for the *QA*Load product screens, as well as detailed step-by-step procedures for all steps in the load testing process and reference information for commands that can be used in *QA*Load scripts.
- *QACenter Installation and Configuration Guide* details the system requirements and procedures for installing each product on the *QA*Center Performance Edition CD-ROM.
- The *Distributed License Management Installation Guide* includes instructions for licensing your *QA*Center products.
- *Release Notes* detail system requirements for using *QA*Load, enhancements made to the product for this release, technical information that may affect how you use the product, any known issues related to using the product, and technical support contact information.

Viewing and Printing Online Books

*QA*Load's online books are provided in PDF format, so you need Adobe Acrobat Reader 3.0 or above to view them. To install the Adobe Acrobat Reader, click **Install Adobe Acrobat Reader** on the *QA*Center CD, or go to Adobe's web site at www.adobe.com.

You can access the online books from the documentation bookshelf. To do so, click the taskbar's **Start** button and choose **Programs>Compuware>QALoad>Bookshelf**. Select the book you want to view.

Because PDF is based on PostScript, a PostScript printer is the most reliable way to print the online books. In most cases, you can also print PDF files to PCL printers. If you cannot print the PDF files to your printer, refer to Adobe's web site at www.adobe.com for troubleshooting information.

Typographical Conventions

The *QA*Load documentation set uses the following typographical conventions:

Description	Examples
Window controls (buttons, menu items, etc.) are shown in **bold type**.	Click **OK**. Select **File>New**.
A fixed pitch font is used for script examples and error messages.	`BEGIN_TRANSACTION();`
Items in angle brackets indicate placeholders for information you supply.	`<userid>, <password>`

World Wide Web Information

To access Compuware Corporation's site on the Internet World Wide Web, point your browser at http://www.compuware.com. The Compuware site provides a variety of product and support information.

FrontLine Support Web Site

You can access online technical support for Compuware products via our FrontLine support web site. FrontLine provides you with fast access to critical information about your *QA*Center product. You can read or download documentation, frequently asked questions, and product fixes, or directly e-mail Compuware with questions or comments.

In order to access FrontLine, you must first register and obtain a password. To register, point your browser at http://frontline.compuware.com. FrontLine is currently available for customers in the United States and Canada. FrontLine services for other countries will be available in the future.

Getting Help

At Compuware, we strive to make our products and documentation the best in the industry. Feedback from our customers helps us maintain our quality standards. If you need support services, please obtain the following information before calling Compuware's 24-hour product support hotline:

- The release (version), and build number of your *QA*Load product. This information is displayed when you select the About command from the Help menu. The name and release are also on the covers of the product documentation.
- Installation information, including installed options and whether it is installed in the default directories.
- Environment information, such as the operating system and release on which the product is installed, memory, hardware/network specifications, and the names and releases of other applications that were running.
- The location of the problem in the *QA*Load product software, and the actions taken before the problem occurred.
- The exact product error message, if any.
- The exact application, licensing, or operating system error messages, if any.
- Your Compuware client, office, or site number, if available.

***QA*Load Customer Support**
Compuware Corporation
One Campus Martius
Detroit, MI 48226-5099
1-800-538-7822

Chapter 1. *QA*Load Testing Overview

Load testing of server-based applications need not involve complete simulation of user activity. Many distributed applications support different types of clients using standard communication methods. *QA*Load allows you to load test servers by sending high-level requests directly from *QA*Load to the server. This process eliminates the need to emulate a specific client platform.

*QA*Load testing duplicates clients interacting with the server portions of a client/server application. *QA*Load emulates users by driving the servers under test in the same way your actual clients do. For example, if your production clients connect to a database server using TCP/IP over Ethernet, *QA*Load makes the same database requests as the workstations. Unlike testing with real machines, you do not need the hardware or staff to create full test scenarios.

*QA*Load tests emulate terminal, browser, or client-application requests through user-developed test scripts. The scripts consist of *QA*Load test management commands and standard C commands, which invoke processing on your servers. *QA*Load allows you to scale up a load on a host or server by running multiple copies of scripts from one or more workstations at the same time. This saves you staffing and investment when conducting large-scale stress tests.

In addition to creating predictable and repeatable loads, *QA*Load can measure host or server response times from the client's perspective. Typically, you want to measure the time that the host or server takes to complete client requests such as database inquiries, WWW requests, and Tuxedo transactions. *QA*Load centrally records this information for subsequent analysis. *QA*Load duplicates and measures the load of handling large numbers of clients and their associated networking functions.

Introduction to 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 *QA*Load scripts by creating the same types of requests that your applications invoke on the server. Each transaction becomes its own script. The *QA*Load 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 *QA*Load Player component, you can specify transaction rates as well as fixed and random delays to better emulate real-system activity. *QA*Load 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. *QA*Load'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 *QA*Load's Analyze component to create presentation-quality reports and graphs.

Script Development Components

The *QA*Load Script Development Workbench is used to develop test scripts. This section provides a brief description of its script development components.

*QA*Load Script Development Workbench

You use the *QA*Load Script Development Workbench to develop 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 *QA*Load's Conductor and Player components to test your system.

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

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

Test Components

As shown in Figure 1-1, a typical load test setup consists of a *QA*Load Conductor, one or more *QA*Load Players, and the system under test. This section provides an overview of these test components.

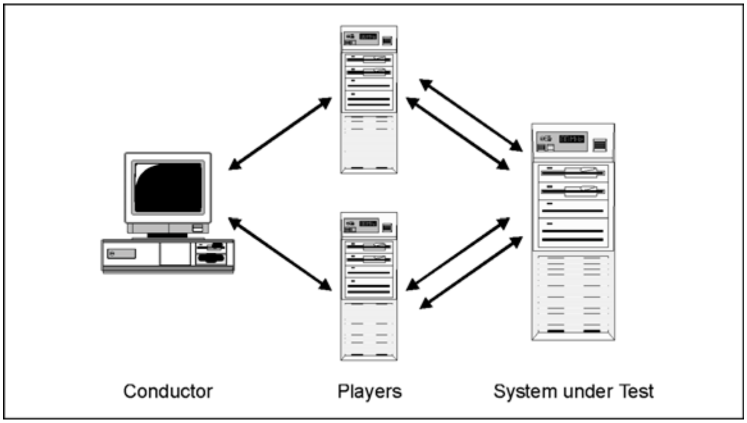


Figure 1-1. Test Components

*QA*Load Conductor

*QA*Load'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.

*QA*Load Player

A *QA*Load Player creates virtual users that simulate multiple clients sending middleware calls to a server under test. In a typical test setup, one or more *QA*Load Player workstations run under any Windows 32-bit platform (Windows XP/2000/NT) or UNIX. For large tests (thousands of simulated clients), you can connect a maximum of 32 Players to *QA*Load'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 *QA*Load Player. Therefore, a Windows Player may simulate 50 to 100 users; single-processor RISC systems may simulate 200-400 users, and large multiprocessor systems may simulate in excess of 1,000 users. The maximum number of virtual users per Player is 1024.

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.

Performing Load Tests Using *QA*Run Scripts

*QA*Load provides you with the functionality to perform load tests using your *QA*Run scripts. By inserting your *QA*Run script into a *QA*Load script template, you can time your GUI-driven business transactions and include those timings in *QA*Load post-test reports. For information on preparing a *QA*Run script for load testing, refer to Chapter 4, "Including QARun Scripts in Load Tests".

Server and Performance Monitoring

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

This section briefly describes each method, and explains where to find detailed information to help set up a test to include the appropriate method.

Remote Monitoring

Remote Monitoring is *QA*Load's own solution that allows you to monitor Windows Registry counters or SNMP counters from a remote machine without installing any software on the remote machine.

To use Remote Monitoring:

- To monitor Windows Registry counters, you must have login access to the machines you want to monitor.
- You must select the counters to monitor on the machines to monitor using the *QA*Load Conductor's Machine Configuration tab.
- To collect SNMP counters, SNMP must be enabled on the remote machine being monitored. Refer to your operating system help for information about enabling SNMP.

While your test is running, *QA*Load will collect the appropriate counter data and write it to your timing file where you can view it in Analyze after the test.

To configure a test to use Remote Monitoring, refer to the *QA*Load Online Help while setting up a test. Online help can be accessed directly from any *QA*Load menu, or by pressing F1 from any *QA*Load screen.

Server Analysis Agents

Server Analysis Agents use enhanced EcoTOOLS 7.2 technology to provide server utilization data without a complete EcoTOOLS deployment. Server Analysis Agents can be installed from the *QA*Center Performance Edition CD onto 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.

Server Analysis Agents are configured from the the *QA*Load Conductor. Data collected through the Agents is automatically downloaded to the *QA*Load Conductor and is available for post-test analysis through *QA*Load Analyze.

To configure a test to use Server Analysis Agents, refer to the *QA*Load Online Help while setting up a test. Online help can be accessed directly from any *QA*Load menu, or by pressing F1 from any *QA*Load screen.

Application Expert\Application Vantage

Application Expert is a Windows-based tool that enables you to examine the effects the network will have on the performance of new or modified applications prior to live deployment. Application Expert provides granular thread details that allow network managers to identify poorly performing applications. *QA*Load integrates with Application Expert to help you analyze network performance during a load test, and provides test data that you can open in both Application Expert and Application Vantage.

Before *QA*Load can collect network data during a load test, the following must be true:

- The Vantage Agent is installed on the same machine as the *QA*Load Conductor. If Application Vantage is installed on the *QA*Load Conductor machine, you can install either the Vantage Agent or the Application Vantage Remote Agent.
- The *QA*Load Conductor and *QA*Load Player machines are located on the same LAN.
- You have set up IP address pairs to monitor in the *QA*Load Conductor.

To configure a test to integrate with Application Expert, refer to the *QA*Load Online Help while setting up a test. Online help can be accessed directly from any *QA*Load menu, or by pressing F1 from any *QA*Load screen.

ServerVantage

ServerVantage (formerly EcoTOOLS) monitors the availability and performance of applications, databases and servers, allowing users to centrally manage events across all application components- Web servers, firewalls, application servers, file systems, databases, middleware, and operating systems. ServerVantage simultaneously monitors these components, analyzes both historical and real-time events, and correlates monitored information for problem detection.

Integration with ServerVantage is configured from the *QA*Load Conductor. Performance counters collected during a load test are included in the test’s timing file and can be sorted and displayed in *QA*Load Analyze in much the same way as *QA*Load timing data.

To configure a test to integrate with ServerVantage, refer to the *QA*Load Online Help while setting up a test. Online help can be accessed directly from any *QA*Load menu, or by pressing F1 from any *QA*Load screen.

EcoTOOLS 6 (UNIX only)

EcoTOOLS contains hundreds of Agents that can monitor resources on your servers. Specifically, *QA*Load supports Agents for monitoring UNIX systems. By using EcoTOOLS with *QA*Load, you can generate a load on your UNIX system while monitoring it for resource utilization problems.

*QA*Load’s EcoTOOLS integration allows you to view EcoTOOLS resource utilization data in graphs you create using *QA*Load’s Analyze component. Once you set up EcoTOOLS to monitor your server, it periodically collects resource utilization data. When you execute a load test, *QA*Load synchronizes with EcoTOOLS and runs the test. Upon completion of the test, *QA*Load requests the resource utilization data that EcoTOOLS collected during the test. You can view this data in *QA*Load Analyze in much the same way as *QA*Load timing data.

To configure a test to integrate with EcoTOOLS 6, refer to the *QA*Load Online Help while setting up a test. Online help can be accessed directly from any *QA*Load menu, or by pressing F1 from any *QA*Load screen.

Chapter 2. Getting Started

The purpose of this chapter is to help you prepare to run a test by setting up the Conductor, a test session, and by validating your test scripts.

This chapter includes the following sections:

- **Setting Up the Conductor —** Describes how to start and configure the *QA*Load Conductor.
- **Setting Up a Test Session —** Describes how to set up a test session.
- **Validating Scripts —** Describes how to validate a script by running it in a simple test.

Setting Up the Conductor

This section describes how to start the Conductor, configure the Conductor, and set up a session ID file.

Starting the Conductor

1. Start the Conductor:

   From Windows:

   — From the Windows taskbar, click the **Start** button.
   — Select **Programs>Compuware**>**QALoad>Conductor**.

From a Command Prompt:

— Type **mpwin32 <session_file_name> /l /e /t**

The applicable parameters are defined in Table 2-1.

Table 2-1. *QA*Load Conductor Command Line Parameters

Parameter	Definition
/l (Optional)	Creates a log file showing error messages and test status.
/e (Optional)	Executes the indicated session ID file. If you don't specify a session ID file, a new session will open.
/t (Optional)	Executes Conductor at a set time. Valid time formats are /txx:xx or /txx/xx/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 shown in Figure 2-1 below. For information on the Conductor's configuration and setup toolbar buttons, refer to the Conductor's online help.

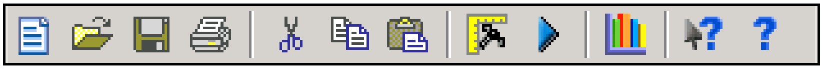
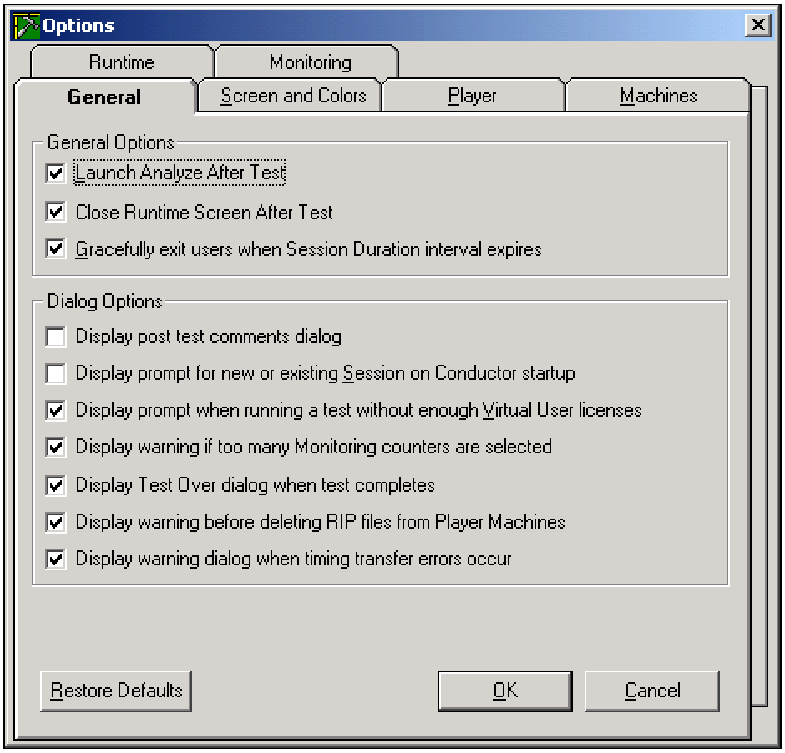


Figure 2-1. Conductor Configuration and Setup Toolbar

Configuring the Conductor

After starting the Conductor, you may need to verify that the Conductor's configuration parameters are set properly using the following procedure:

1. Click the **Options** button or select **Tools>Options**. The Options dialog box opens.



General Options

☑ Launch Analyze After Test

☑ Close Runtime Screen After Test

☑ Gracefully exit users when Session Duration interval expires

Dialog Options

☐ Display post test comments dialog

☐ Display prompt for new or existing Session on Conductor startup

☑ Display prompt when running a test without enough Virtual User licenses

☑ Display warning if too many Monitoring counters are selected

☑ Display Test Over dialog when test completes

☑ Display warning before deleting RIP files from Player Machines

☑ Display warning dialog when timing transfer errors occur

2. On the **General** tab:

   a. In the **General Options** area:

      — Select **Launch Analyze After Test** to start *QA*Load's Analyze program immediately after the test completes.

      — Select **Close Runtime Screen After Test** to automatically close the test window after a test has finished.

      — Select **Gracefully exit users when Session Duration interval expires** to gracefully exit all virtual users when the test duration you specified on the Test Information tab runs out.

   b. In the **Dialog Options** area, determine which warnings and prompts *QA*Load Conductor should display. For details about each, press F1 from the **General Options** tab to access online help.

3. On the **Screen and Colors** tab:

   a. Use the **Color Settings** area to change the colors of *QA*Load Conductor rows and headers, if necessary.

   b. Select the **Enable Row Divider Lines** check box to enable grid lines on the Test Information Window for better visual separation of data.

  c. In the **Refresh Intervals** area, use the slider controls to set the rates at which the *QA*Load Conductor interface, runtime graph, and Summary Report are refreshed at runtime.

  d. In the **Show Timings** area, select how to display timings in the Runtime Window while a test is running: the average, minimum, maximum, or the current timing.

4. On the **Player** tab:

   a. Select the **Show Communications** check box to display all messages from the Conductor to the Player on the Player's Main Window at runtime.

   b. Select the **Log Messages to File** check box to record all messages from the Conductor to the Player in a text file in your installation directory. The file will be named Player####Messages.log, where #### is the Player's port number.

   c. Select the **Start Player Minimized** check box to launch the *QA*Load Player minimized on your task bar when starting a test.

   d. Random number generation is used within the Player to determine end of transaction (End_Transaction) and Rnd_delay sleeps. You may also use random number generation in your script.

      The Conductor allows you to determine whether to use the system-generated sequence of numbers, or whether to seed the sequence so that virtual users use different sequences.

      — Choose **All users the same** to assign the same random number sequence to all virtual users.

      — Choose **Different for each user** and then enter values in the **Seed** and **Multiplier** fields to set a different random number seed for each user. The seed value is determined by the following algorithm:

        seed value = <seed> + <virtual user #> * <Multiplier>

        The <virtual user #> is a value ranging from 0 to N-1, where N is the number of simulated users.

   e. In the **Display Options** area, determine how many lines of messages from the Conductor should be displayed on the Player's Main Window at runtime. The default is 2000.

   f. In the **VU Startup Delay** field, set a startup interval (in milliseconds) for virtual users.

5. On the **Machines** tab:

   a. In the **General Options** area:

      — Select **Require requesting before allowing assignment** if a Player machine should be successfully requested before it can be added to a test.

- — Select **Query network name during discover machines** if you want the Conductor to return network names instead of IP addresses for each workstation where the *QA*Load Player is installed.
- — (WWW tests only) Select **Generate IP Spoofing Data** for the Conductor to command the Players to create local datapools of static IP addresses at runtime to direct each virtual user on that machine to use a different source IP address.
- — Select **Show warning for machines with incompatible versions** to be prompted when the *QA*Load Conductor discovers Player machines on the network that have previous version of *QA*Load installed.

b. The **Discovery Options** area lists the port number(s) the Conductor should monitor when discovering available Players. The default port is 3031. If you changed the port number of any Player, add it to the Discovery Options area by typing it in the edit field and clicking the Add button.

c. In the **File Transfer Options** area, indicate when/if to automatically transfer the script from the Conductor to the Players, and whether to transfer Java information.

d. The **Machine Defaults** area displays the recommended number of thread- and process-based Players to assign per 64 MB of RAM, and the default port number for the first Player launched. You can change these values, if necessary.

6. On the **Runtime** tab:

   a. In the **Timing Information Transfer Rate** area, select whether to disable timing transers, always send all timing information including checkpoint timings, or send summary information at intervals you specify. If you select Send Periodic Updates containing Summary information, use the slider control to set a time interval, in seconds, at which to transmit data.

   b. Select the **Delete timing file after successful transfer** check box to delete the timing file after it has been transferred successfully to the Conductor from the Players. Then, set a value at which the Conductor should stop attempting to transfer the file if the transfer is not successful.

   c. In the **Counter Data** area, select **Enable Counters** to track Player counters at test time. That is, how many virtual users are running and how many have failed. Then choose whether to graph that data at runtime (Retrieve at runtime) or simply store it in the timing file for later analysis (Store in timing file only).

7. On the **Monitoring** tab, configure the intervals at which monitoring information is gathered by Server Analysis Agents or Remote Monitoring Agents, or for top processes.

8. When you are finished, click **OK**.

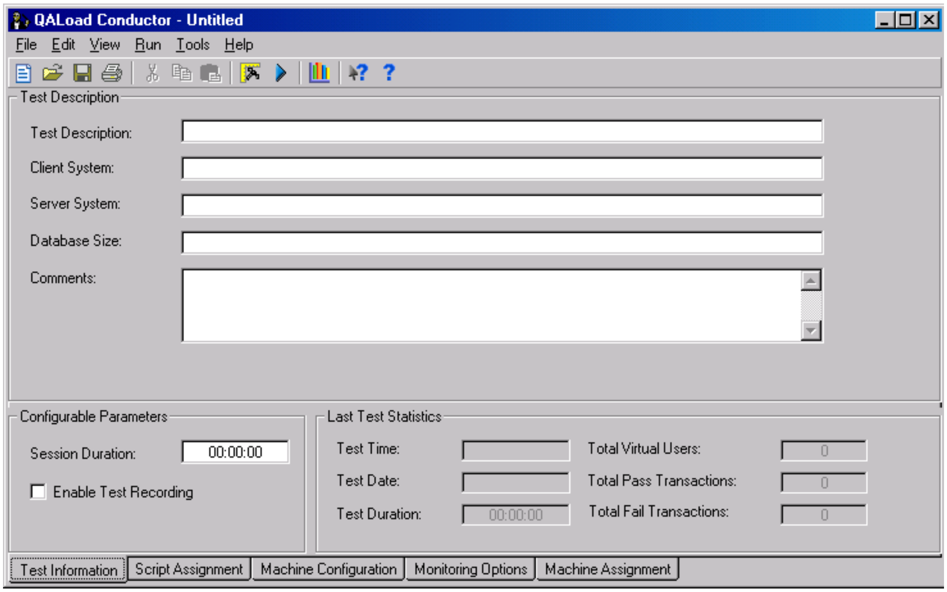
For more information on the Options dialog box, refer to the Options dialog box topic in the *QA*Load online help.

Setting Up a Test Session

For every test you run you will 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 directory \QAload\Session.

To define a session ID file for your test, complete the following steps.

1. From the Conductor's **File** menu, select **New** to open a new instance of the Test Information Window.



Test Description:
Client System:
Server System:
Database Size:
Comments:

Session Duration: 00:00:00
□ Enable Test Recording

Test Time:		Total Virtual Users:	0
Test Date:		Total Pass Transactions:	0
Test Duration:	00:00:00	Total Fail Transactions:	0

2. On the **Test Information** tab:

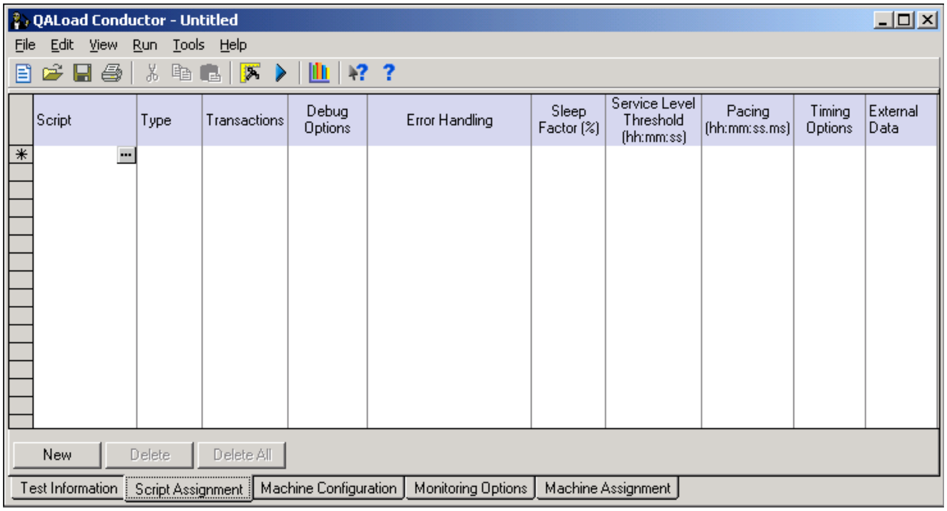
   a. Enter descriptive information about the test in the **Test Description**, **Client System**, **Server System**, **Database Size**, and **Comments** fields.

   b. In the **Session Duration** field, enter a value to specify the maximum duration for the test to run. Enter zero if you do not want to specify a maximum duration.

   c. Select the **Enable Test Recording** check box to save a visual re-enactment of the load test.

   d. In the **Time Out** field, enter a value to specify the amount of time a virtual user waits for a response. If the virtual user does not get a response within the specified time period, the virtual user displays a time out status.

3. On the **Script Assignment** tab, add the appropriate scripts to the test and set script options for each:



   a. Click the **New** button 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 from the Scripts of Type drop-down list.

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

   c. (SAPGUI only) Click the browse [**...**] button to set custom options for the middleware.

   d. In the **Transactions** column, specify the maximum number of transactions that you want each virtual user running this script to 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

   e. In the **Debug Options** column, click the browse button to open the Debug Options dialog box, where you can specify virtual users to display trace information or generate log files for debugging.

   f. In the **Error Handling** column, choose whether to restart, continue, or abort a transaction upon encountering a non-fatal error.

   g. 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 sleep).

h. 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.

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

j. In the **Timing Options** column, click the browse button to open the Timing Options dialog box, where you can set options related to checkpoints and timing data thinning. Press F1 from the Timing Options dialog box for descriptions of the available options.

k. (Optional) In the **External Data** column, click the browse button to open the External Data dialog box to associate any external files with your selected script. Press F1 from the External Data dialog box for descriptions of the available options.

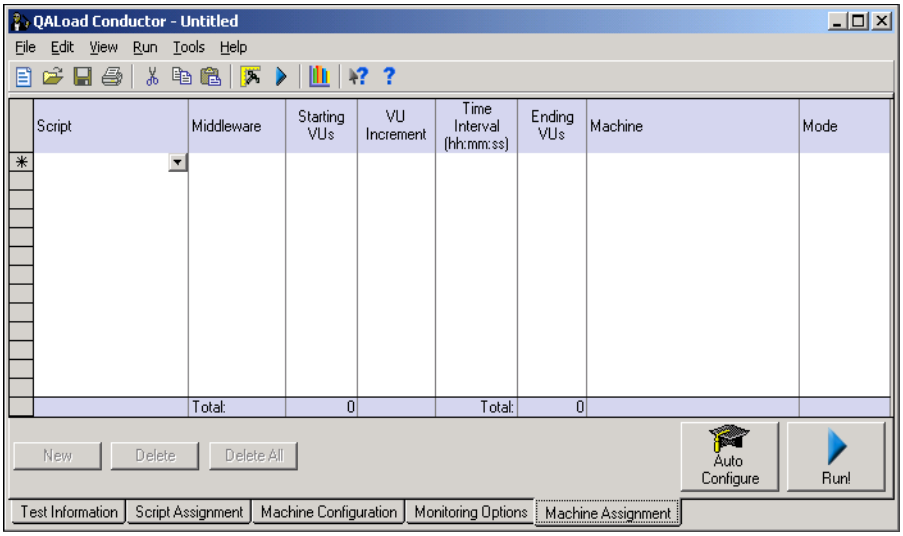
— In the **Central Datapool** area, click the browse button to navigate to a datapool file to use. Then, select the **Rewind** check box if you want to re-use the datapool data in this test once the last field in the file has been used, or select the **Strip** check box to remove the datapool from the script after using it, that is, if you do not want to re-use the same set of data from your datapool in subsequent tests.

— In the **Attached Files** area, click the **Add** button to navigate to any additional files you may need to include in your test. For instance if you are posting binary files to a web site, those files can be downloaded to the Players at runtime if you add them.

4. On the **Machine Configuration** tab, configure Player Agent workstations, Server Analysis Agents, Remote Monitor Machines, and integration with Application Expert. Select the appropriate option to configure each in turn. Press F1 from the Machine Configuration tab for descriptions of each option to set.

5. (Optional) Use the **Monitoring Options** tab to set options for your integration with Compuware's EcoTOOLS 6 or ServerVantage. Press F1 from the Monitoring Options tab for detailed descriptions of the integration options.

6. Click the **Machine Assignment** tab.



Script	Middleware	Starting VUs	VU Increment	Time Interval (hh:mm:ss)	Ending VUs	Machine	Mode
	Total:	0		Total:	0		

7. On the **Machine Assignment** tab:

   a. Assign scripts to specific Player Machines:

   - **Manual assignment** — In the **Machine** column for the appropriate 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 Machine Configuration tab to set up a Player.
   - **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. *QA*Load will automatically assign scripts to each Player machine.

   b. Type the number of virtual users that should start and end the test for each script in the Starting VUs and Ending VUs columns.

   - **Ramp-up scripts** — If *QA*Load should add virtual users incrementally, type a value in the VU Increment column to indicate how many to add with each increment. Then, determine how often *QA*Load should increment virtual users in the Time Interval column. Make sure the Ending VUs column is larger than the Starting VUs column.
   - **Dial-up\Dial-down scripts** — To add (dial-up) or remove (dial-down) scripts or virtual users at runtime manually, type values in the Starting VUs and Ending VUs columns but leave the VU increment and Time Interval columns blank. At runtime, use the dial control in the Test Control Panel to dial-up or dial-down the number of virtual users running.

   When all scripts have been assigned and configured correctly, the Run button on the Machine Assignment tab will become available.

8. When you are finished, choose **File>Save As** to save the session ID file with a unique name.

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 Win32 scripts from the *QA*Load Script Development Workbench, the *QA*Load Player, or the *QA*Load Conductor. All three methods are described in the sections that follow. Use the method that is most convenient for you.

This section describes how to validate Win32 and UNIX C-based scripts by running them in a simple test.

Note

(WWW scripts only) Before you run a UNIX script containing SSL requests, copy the file do_www.exp (for 40-bit encryption) or do_www.128 (for 128-bit encryption) from your *QA*Load operating system-specific UNIX directory (for example, c:\program files\compuware\qaload\unix\solaris) to the UNIX directory where the *QA*Load Player is installed. Then, rename the file do_www.so.

Validating from the *QA*Load Script Development Workbench

1. With a session open in the *QA*Load Script Development Workbench, select **Options>Workbench** to configure the *QA*Load Script Development Workbench and Player for validation.
2. On the **Script Validation** tab:
   a. Select the **Automatically Recompile** check box if you want *QA*Load to compile a script before attempting to validate it. *QA*Load will list any compilation errors in the editor after compiling.

   b. (Java or OFS only) Select the **Ask for Automatic Compilation of Java and OFS Scripts** check box to always be prompted to validate a Java or OFS script after compiling it.

   c. Select the **Only Display Player Output on Script Failure** check box to only generate a Validation error report (.val) if the script fails.

d. Type a value in the **Wait up to** field the *QA*Load Script Development Workbench should wait for a script to execute before timing out.

Note

For SAP and Citrix, due to the time required to logon to the server, you may need to increase the timeout value to 100 seconds or more, depending on your particular setup. Set the timeout value to 100 seconds or to the length of the capture (in seconds), whichever is greater.

e. In the **Player Settings** area, select the **Abort on Error** check box for *QA*Load to stop script execution upon encountering an error.

f. Select the **Debug Data** check box for the script to display a debug message indicating which command the script is executing.

g. In the **Run As** area, indicate whether the transaction should be run as thread- or process-based.

h. Type a number of virtual users to assign for validation in the **Number of users** field.

i. Type the number of transactions to run in the **Transactions** field. For validation, Compuware recommends that you accept the default value of 1 transaction.

j. In the **Sleep Factor %** field, type a value to indicate the percentage of an originally recorded sleep (pause) to maintain when validating a script.

3. Click **OK** to save your changes.
4. In the Workspace window, click the **Scripts** tab to access the available scripts.
5. Double-click on the appropriate script name to open the script.
6. From the **Session** menu, select **Validate Script**.

If the script executes successfully, it is valid to use in a load test. If it does not, a Validation error report (.val) will open in the Output window to help you identify errors.

Validating from the *QA*Load Player

1. From the **Compiled Script** field, browse for the compiled script DLL you want to validate. Compiled scripts are usually located in the directory \Program Files\Compuware\QALoad\Scripts.
2. Assign a number of virtual users to validate the script in the **Users** field. Compuware recommends one user for script validation.
3. Assign a number of transactions in the **Transactions** field. Compuware recommends one transaction for script validation.

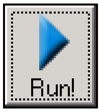
4. Select the **Debug Data** check box to have the Player display a debug message indicating each command as it is executed.
5. Select the **Abort on Error** check box if you want the *QA*Load Player to stop script execution upon encountering an error in the script.
6. In the **Run As** area, select whether the transaction should run as thread- or process-based.
7. Click the **Start** button to run the script.

If the script runs without errors, it is valid to use in a load test.

Validating from the *QA*Load Conductor

You can run the Conductor and Player on separate machines or on the same machine to validate your Win32 script.

1. From the *QA*Load Conductor, click the **Script Assignment** tab.
2. Click the **New** button to access the Select Script dialog box.
3. Highlight the script you wish to use and click the **Select** button to return to the **Script Assignment** tab.
4. Type a **1** (one) in the **Transactions** column to indicate that you want the virtual user running this script to run one transaction.
5. Click the **Machine Assignment** tab.
6. Type a **1** (one) in the **Starting VU's** column for the script to assign it to 1 virtual user.
7. Select your Player machine from the **Machine** drop-down list. If the appropriate Player machine is not listed, click the Machine Configuration tab to set up a Player machine.



8. To start the test, click the **Run** button on the **Machine Assignment** tab or choose **Run>Start Test**. You can monitor the progress of your script's execution by reading the debug messages on the Player machine.

   In addition, you can use the Conductor debug function to see the script executing commands. To start the Conductor debug function, click the **Debug** button on the Conductor's Runtime toolbar.

If the script runs to the end without any errors, it is valid to use in a load test.

Chapter 3. Running a Load Test

After validating a script using one of the methods described in “Validating Scripts” on page 2-10, it is safe to run a load test with that script. This chapter describes how to prepare for, start, monitor, and stop a load test. It includes the following sections:

- **Preparing for a Load Test —** Describes the tasks to complete before running a load test.
- **Starting a Load Test —** Describes how to start a test.
- **Monitoring a Load Test —** Describes how to monitor a test, including Player activity and datapool usage.
- **Stopping a Load Test —** Describes how to stop a load test and generate a timing file.
- **Viewing SAP 4.x Post-Test Log Files —** Describes how SAP testers using versions 4.x can view detailed information about each request and response.

If the test scripts you intend to use have not been validated to ensure they run correctly, see “Chapter 2, Getting Started” before completing the procedures in this chapter.

Preparing for a Load Test

Before you run a load test, you need to 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 on datapool files, see “Chapter 4, General Advanced Scripting Techniques” in the *QALoad Script Development Guide*.

- **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, refer to the *QALoad Script Development Guide*.

  Once you export or create the necessary Client Certificates, you can insert them into your script using data pools. For more information on using datapools, refer to the *QALoad Script Development Guide*.

To start a load test, see "Starting a Load Test" on page 3-2.

To run a batch test (a series of tests) see "Running a Batch Test" on page 3-7.

Starting a Load Test

Run!

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

While a load test is running, the Conductor's toolbar changes from the Configuration and Setup Toolbar to the Runtime Toolbar shown in Figure 3-1. 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 *QA*Load Conductor while a test is running — including descriptions of the Runtime Toolbar buttons — see "Monitoring a Load Test" on page 3-3.

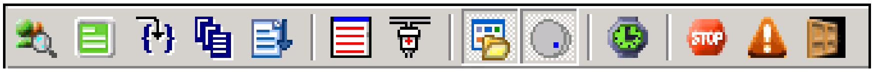


Figure 3-1. Conductor Runtime Toolbar

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

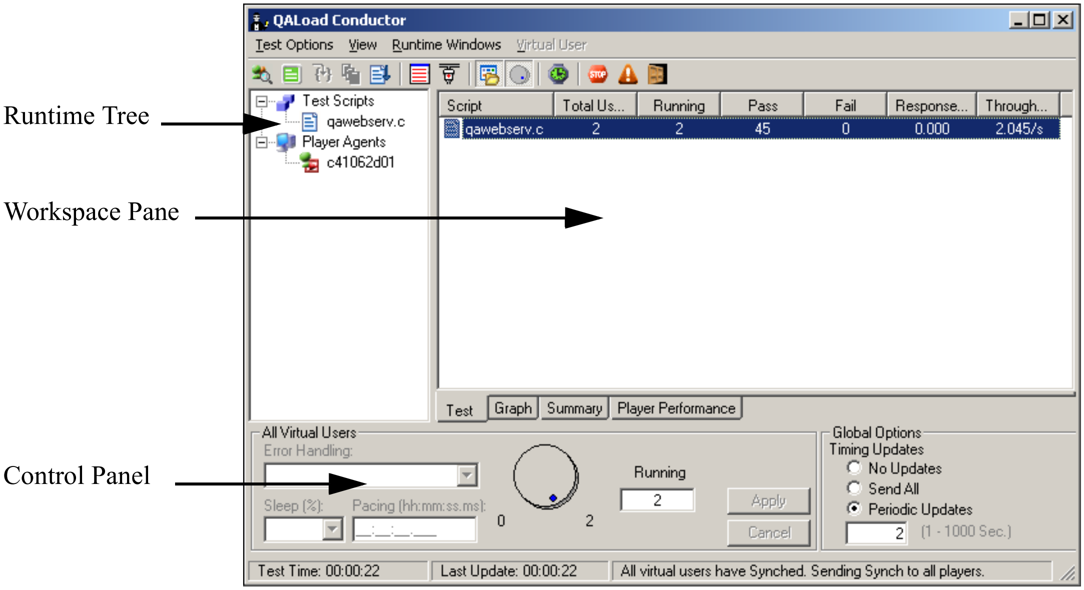
Note

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.

When a test is started, the *QA*Load Conductor's interface changes to an interactive test control station (see Figure 3-2), 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, 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. The following sections briefly describe the Runtime Window and the options available for monitoring a running test. For detailed information, refer to the *QA*Load online help.

Viewing General Test Information

The Runtime window is divided into three panes.



Script	Total Us...	Running	Pass	Fail	Response...	Through...
qawebserv.c	2	2	45	0	0.000	2.045/s

Running: 2

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

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

Figure 3-2. Runtime Window

Runtime Tree

The Runtime Tree displays an interactive tree-view of the scripts and Player machines involved in the test, and displays the status of each. It provides access to real-time statistics and additional views of real-time test data. For instance, you can monitor a script's datapool usage by clicking on the script name and then clicking the Datapool Monitor button.

Workspace Pane

The Workspace Pane opens multiple tabs that display real-time test data, statistics, WWW.rip files, and graphs. The data on these tabs is updated according to the time interval you set in your Session ID file before the test on the *QA*Load Conductor's Options dialog box, Runtime tab. You can override those options at runtime using the Global Options area in the bottom tab.

Test Control Panel

The Control Panel is a dockable control station that allows you to change virtual user options and data transfer options while your test is running. The Control Panel is the means of controlling dial-up\dial-down scripts, which allow you to manually start or suspend virtual users at runtime.

Accessing Detailed Test Data

In addition to the test data shown by default on the Runtime Window, you can access detailed test information using the *QA*Load 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 Control Panel
- synchronize all virtual users
- exit, abort, or quit the test

These options are briefly described in the following sections. For more detailed information, refer to the *QA*Load Online Help.

View Virtual User Statistics

Click the Virtual User Info button to open the Virtual User Info dialog box, which displays detailed information about the selected virtual user.

(WWW) View Virtual User Activities

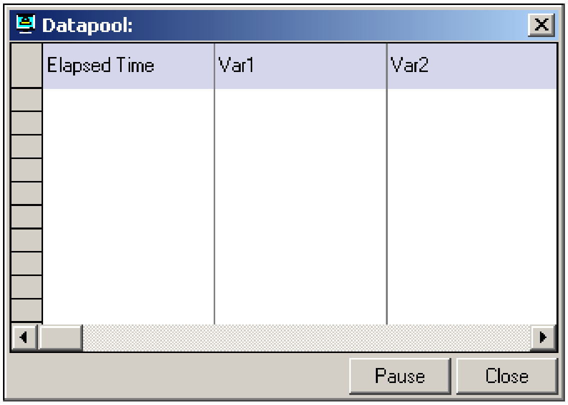
(WWW only) Click the Monitor Web User button to open the Web User Monitor in the top pane, displaying the selected virtual user's script in a browser window.

(WWW) Step to the Next Request

(WWW) Click the Step button to step to the next request in the Web User Monitor.

Display the Current Datapool Record

Click the Datapool Monitor button to display the current datapool record the script is using.

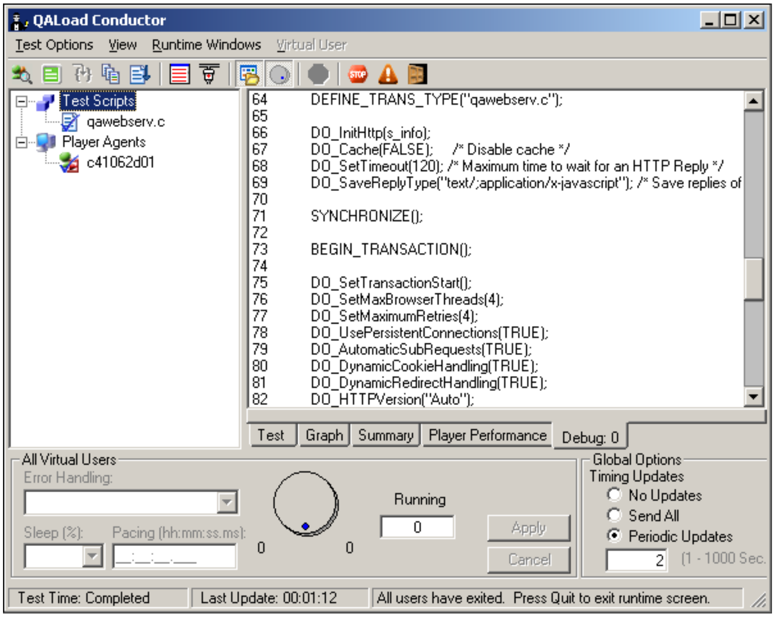


Elapsed Time	Var1	Var2

Figure 3-3. Datapool Window

Display a Virtual User's Running Script

Click on a Player in the tree-view and then click the Debug tab button to open a tab in the Workspace pane displaying the script's statements as they are executed (see Figure 3-4). Note that the Conductor highlights the line in the script that is currently executing.



```
DEFINE_TRANS_TYPE("qawebserv.c");

DO_InitHttp(s_info);
DO_Cache(FALSE);    /* Disable cache */
DO_SetTimeout(120); /* Maximum time to wait for an HTTP Reply */
DO_SaveReplyType("text/;application/x-javascript"); /* Save replies of

SYNCHRONIZE();

BEGIN_TRANSACTION();

DO_SetTransactionStart();
DO_SetMaxBrowserThreads(4);
DO_SetMaximumRetries(4);
DO_UsePersistentConnections(TRUE);
DO_AutomaticSubRequests(TRUE);
DO_DynamicCookieHandling(TRUE);
DO_DynamicRedirectHandling(TRUE);
DO_HTTPVersion("Auto");
```

Running: 0

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

Test Time: Completed | Last Update: 00:01:12 | All users have exited. Press Quit to exit runtime screen.

Figure 3-4. Debug Window

Display Player Messages

Select a Player in the tree-view and then click the Display Player Messages button to toggle the Player Messages window, which displays any messages sent from the Player to the *QA*Load Conductor at runtime.

Display Communication Statistics

Select a Player workstation in the tree-view and then click the Show messaging statistics button to open the Conductor\Player Message Statistics dialog box, displaying information about the communications between the *QA*Load Conductor and the selected Player workstation at runtime.

Show/Hide Runtime Tree or Control Panel

Click the Show/Hide Runtime Screen Tree button to toggle the tree-view.

Click the Show/Hide Control Panel button to toggle the Test Control Panel.

Synchronize Virtual Users

Click the Synchronize button to manually release all virtual users that are pausing script execution as a result of encountering a SYNCHRONIZE() command.

Exit, Abort, or Quit a Test

Click the Exit button to cleanly stop all running virtual users.

Click the Abort button to force all virtual users to stop executing the script.

Click the Quit button to completely quit the load 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 is comprised of multiple session ID files that are executed sequentially.

Creating a Batch File

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 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 prior to starting the batch test. This means the .c files referenced in the selected session ID files must be present in the scripts directory.

1. From the Conductor menu, select **Run>Batch Test**.
2. Select the session ID files to add from the Session Files list and click the **Add** button. Sessions included in the current batch file are listed in the Batch List.
3. When you are finished adding sessions, click the **Save** button to name and save the batch test. Batch tests are saved in your scripts directory with the file extension .run. For example, c:\Program Files\Compuware\QALoad\Middlewares *Oracle*\Scripts*batch*.run.

Running a Batch Test

1. From the Conductor main menu, select **Run>Batch Test**.
2. If a batch test you already defined is not listed in the Batch List, click the **Load** button and navigate to the appropriate batch test.

   To create a new batch test, select session ID files from the Session Files list and click the Add button to add them to the batch.
3. Set a time delay between tests, in seconds, in the **Delay Between Tests** field.
4. To begin running the test, click the **Run** button. The Conductor executes each of the session ID files in sequence. You can cancel a batch test at any time by clicking the **Cancel** button.

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:

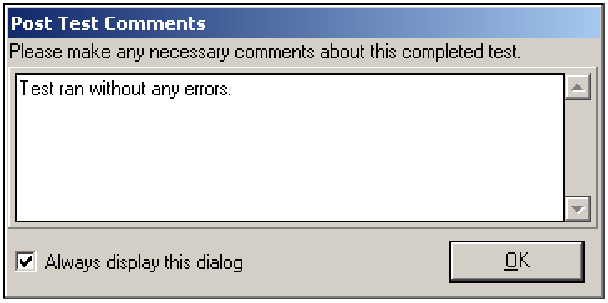
- A script encounters an EXIT command.
- A script completes its transaction loop.

STOP

To stop a load test, click the **Exit** button.

Adding Post Test Comments to the Test

If you selected the Display Post Test Comments dialog option on the General tab of the Options dialog box when you configured the Conductor, the Post Test Comments window in Figure 3-5 opens when you click the **Quit** button. Type any comments, which will then be saved to the test's Summary Report which is viewable in *QA*Load Analyze.



Post Test Comments

Please make any necessary comments about this completed test.

Test ran without any errors.

☑ Always display this dialog

Figure 3-5. Post Test Comments Window

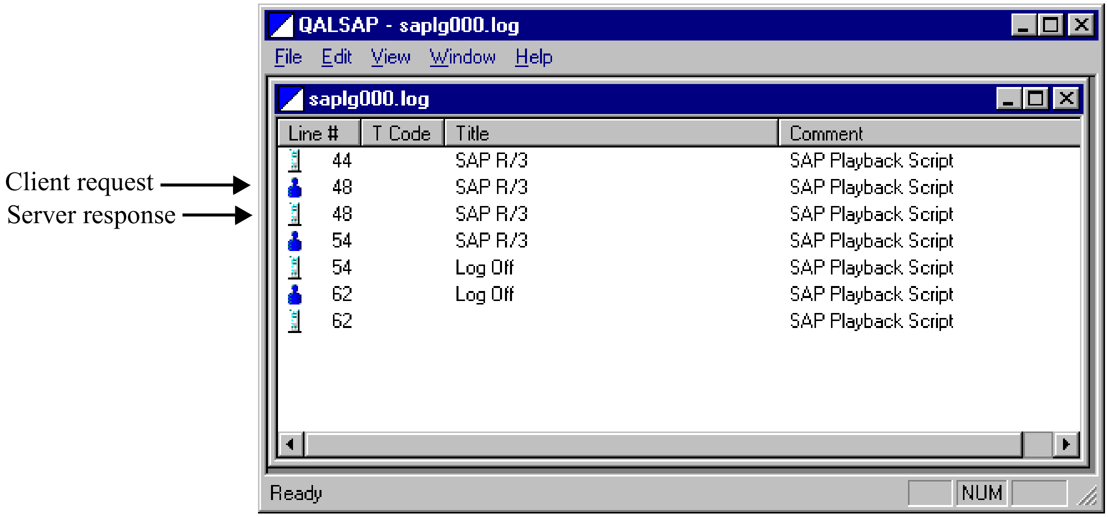
Viewing SAP 4.x Post-Test Log Files

If you selected the option Detailed Logging on the *QA*Load Script Development Workbench's SAP Conversion Options dialog box before you ran the test, *QA*Load automatically generated a log file for each virtual user named `saplg###.log` (where ### is the virtual user number) during the test. Each log file contains a graphical representation of the events sent to and received from the server for a particular virtual user. You can open and delete a log file from the *QA*Load Script Development Workbench using the following procedure.

Opening a Virtual User Log File

1. With a SAP session open in the *QA*Load Script Development Workbench, select **File>Browse**.
2. On the Browse dialog box, double-click Log Files.
3. After *QALSAP* opens, select **File>Open**.
4. In the Files of Type field, select **Log File**. The Browse Log Files dialog box opens, displaying the available SAP log files.
5. Double-click on the log file you wish to open to open. Log files are named `saplg###.log`, where ### is the virtual user number.

QALSAP opens the selected log file. Figure 3-6 shows a virtual user log file open in *QALSAP*. Note that, in the Line # column, each request by the client is marked by a blue "client" icon, while each response from the server is marked by a white "server" icon.



Line #	T Code	Title	Comment
44		SAP R/3	SAP Playback Script
48		SAP R/3	SAP Playback Script
48		SAP R/3	SAP Playback Script
54		SAP R/3	SAP Playback Script
54		Log Off	SAP Playback Script
62		Log Off	SAP Playback Script
62			SAP Playback Script

Figure 3-6. SAP virtual user log file, open in QALSAP

Viewing Request or Response Details

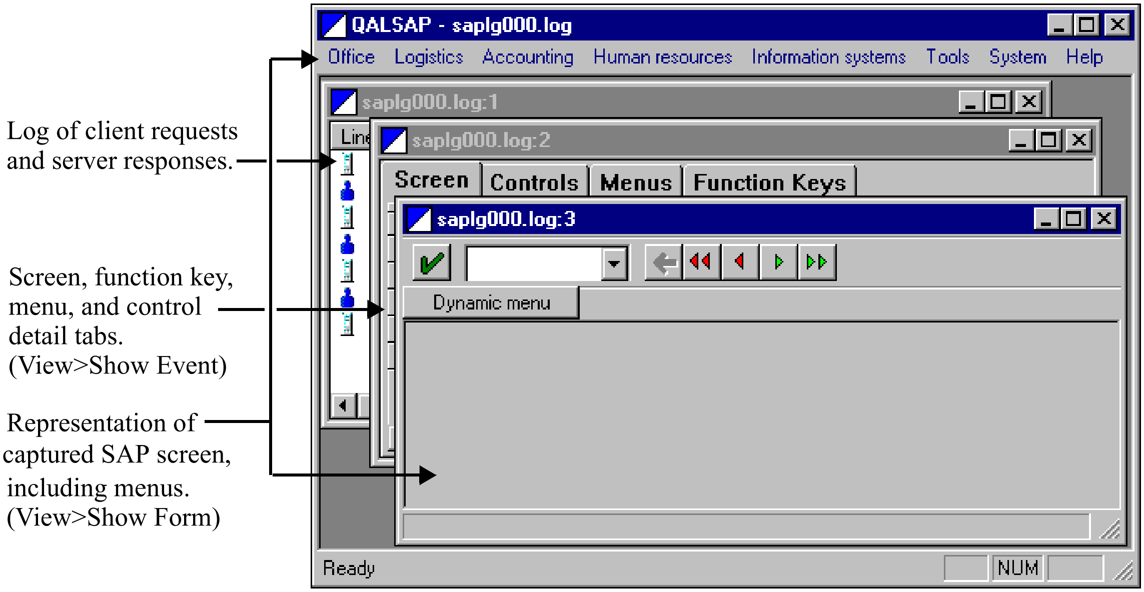
QALSAP allows you to view detailed information about each request and response in the log file, including each logged SAP screen, and its menus, function keys, and controls. You determine which information to display by selecting one or both of the following commands from the View menu:

- **View>Show Event** — Opens a detail window, displaying detailed information about each logged event, including screen names, key names, tool tip text, and so on.
- **View>Show Form** — Opens a form window, displaying a graphic representation of the logged SAP screen and related menus.

To view a specific request or response:

1. With a log file open in *QALSAP*, select the appropriate command(s) from the View menu to determine what amount of detail to view.
2. Double-click on the line number of the request or response you want to view. Detail windows open for the selected request or response, depending on the options you set in the View menu. In Figure 3-7, both Show Event and Show Form were selected.
   - To view the next like response—for example, if you are viewing a client request and want to view the *next* client request—select **View>Goto Next Response**. Alternately, from the form window, click the appropriate toolbar button to view the next or previous response.

- To view the next event in the log, select **View>Goto Next Event**. Alternately, from the form window, click the appropriate toolbar button to view the next or previous event.



Log of client requests and server responses.

Screen, function key, menu, and control detail tabs. (View>Show Event)

Representation of captured SAP screen, including menus. (View>Show Form)

Figure 3-7. QALSAP log views for a single SAP client request

Chapter 4. Including *QA*Run Scripts in Load Tests

The *QA*Load Script Development Workbench provides a C-based template script that you can build upon to create a *QA*Run script within a *QA*Load script. By entering pertinent information about your *QA*Run script into the *QA*Load Script Development Workbench, you can instruct *QA*Load which *QA*Run script to run and what *QA*Run environment to run the script under. At test time, *QA*Load will launch the specified script in *QA*Run so you can time your GUI-driven business transactions and include those timings in *QA*Load post-test reports.

The procedures in this chapter assume you have *QA*Run installed and operational on the machine(s) where you have *QA*Load's Player installed.

Creating a *QA*Run Script

To create a *QA*Run script, you insert any number of *QA*Run transactions (*QA*Run scripts) into a *QA*Load template script accessible from the *QA*Load Script Development Workbench. The template script is a simple *QA*Load script that can be compiled and run; however, it contains *no functionality* until you insert the *QA*Run transactions appropriate for your testing needs. *QA*Load provides two methods for inserting *QA*Run transactions: automated, and manual.

Using the automated method, you enter information in the *QA*Load Script Development Workbench about the *QA*Run transactions to use and then let *QA*Load generate the test script using the information you provided. This method is fast and efficient when you know exactly which *QA*Run scripts to use and where they are located.

The manual method allows you to open a copy of the *QA*Load template script and insert transactions and commands manually. You may want to use this method if you suspect you may need to edit your script while you're creating it.

Automatically Creating a Test Script

Use the following procedure to automatically create a *QA*Run script.

1. From the *QA*Load Script Development Workbench, select **Session>QARun** to start a *QA*Run scripting session.
2. Select **Session>Generate Script**. The Create New *QA*Run Execution Script dialog box opens.
3. In the **Login String** field, select or type a valid username and password to access your installation of *QA*Run.
4. In the **Environment** field, select the appropriate *QA*Run environment.
5. In the ***QA*Run Script** field, enter the name of the *QA*Run transaction to insert, or select it from the drop-down list, which contains a record of the last five *QA*Run script names you entered. Note that you can enter a script name from any database; however, when the test is actually running and *QA*Load invokes *QA*Run, *QA*Run will attempt to retrieve that script from its default database. Therefore, in the *QA*Run program installed on the Player, you should designate a default database that contains the script(s) you want to run.
6. Select the **Automatically Include Checkpoint** check box if you want *QA*Load to automatically insert a checkpoint into the script after this *QA*Run transaction.
7. In the ***QA*Load Script Name** field, enter a name for this *QA*Load script. To write over an existing script, click the **Browse** button to the right of this field and select a script from the list of available scripts.
8. To add additional *QA*Run transactions to this script, click the **Add Script** button and repeat Steps 3–6 for each additional transaction.
9. When you are finished, click the **Create Script** button. The *QA*Load script is saved in the directory \Program Files\Compuware\QALoad\Middlewares \QARun\Scripts, and the script opens in the script editor.
10. To compile the script for testing, select **Session>Compile**.

Before using this script in a load test, you should validate it by running a simple test to ensure that it plays back correctly. For information on validating a script, refer to "Validating Scripts" on page 2-10.

Manually Creating a Test Script

Use the following procedure to manually insert *QA*Run transactions (scripts) into the *QA*Load template script.

1. From the *QA*Load Script Development Workbench, select **Session>QARun** to start a *QA*Run scripting session.
2. Select **Session>New Template** to create a new script from the *QA*Load template script.
3. In the Choose Script Name dialog box, enter a name for the new *QA*Load script and click **OK**. The script is saved in the directory \Program Files\Compuware \QALoad\Middlewares\QARun\Scripts, and the script opens in the script editor.
4. Edit the script as necessary:
   - You can manually enter any transactions or scripting commands directly in the script.
   - You can insert a *QA*Run transaction by positioning the cursor on the appropriate line and selecting **Session>Insert>Transaction**. On the Insert a *QA*Run Transaction dialog box that opens:
     — In the **Login String** field, select or type a valid username and password to access your installation of *QA*Run.
     — In the **Environment** field, select the appropriate *QA*Run environment.
     — In the ***QA*Run Script** field, enter the name of the *QA*Run transaction to insert, or select it from the drop-down list, which contains a record of the last five *QA*Run script names you entered. Note that you can enter a script name from any database; however, when the test is actually running and *QA*Load invokes *QA*Run, *QA*Run will attempt to retrieve that script from its default database. Therefore, in the *QA*Run program installed on the Player, you should designate a default database that contains the script(s) you want to run.
     — When you are finished, click the **Insert** button to insert the script you just created into the *QA*Load script.
5. When you are finished, save any changes.
6. To compile the script for testing, select **Session>Compile**.

Before using this script in a load test, you should validate it by running a simple test to ensure that it plays back correctly. For information on validating a script, refer to “Validating Scripts” on page 2-10.

Chapter 5. Load Testing the Network with NetLoad

NetLoad is *QA*Load's suite of load generation scripts that allows you to simulate load conditions on your network using any of the following protocols:

- FTP
- HTTP
- PING
- LDAP
- POP3
- SMTP
- TCP
- UDP
- MSExchange

Note

To use NetLoad for MSExchange to test on Outlook 2000, you must ensure that CDO support is installed on your workstation before you continue. For instructions, see "Verifying CDO Support for MSExchange" on page 5-9.

NetLoad includes *QA*Load-provided scripts, which you can access from the Conductor to run in a test, for each protocol. You can customize the activity of the script by creating reusable datapools in the *QA*Load Script Development Workbench to use during testing. When you run a test, each virtual user will request a single datapool record. Once all the records have been read, the datapool file is rewound and the process starts again. You can use *QA*Load's components to run scripts and analyze the results as usual, or you can integrate your results with Compuware's EcoTOOLS product.

In short, NetLoad allows you to generate traffic on your network in a controlled manner and gather performance timings from the network. To facilitate testing under TCP/IP and UDP, NetLoad provides you with a server module to simulate server activity — allowing you to gather network timings without expending your actual server resources.

For more information on the NetLoad Server modules, see "NetLoad Server Modules for TCP/IP and UDP" on page 5-2.

To create a datapool to begin testing under any protocol, see "Creating a Datapool" on page 5-4 and any other applicable procedures that follow.

NetLoad Server Modules for TCP/IP and UDP

If you are load testing a network running TCP/IP or UDP, you should use the appropriate NetLoad Server module to simulate server responses during your load test. This allows you to load your network and collect timings without expending your own server's resources. The NetLoad Server modules are *only* for use if you're testing on TCP/IP or UDP. *You do not need to install the Server modules to test any other NetLoad-supported protocol*. If you are not sure if you should use a NetLoad Server module, see "Determining When to Use the TCP Server Module" and "Determining When to Use the UDP Server Module" below.

You can install or copy the NetLoad Server modules to any Windows 98 or later or NT workstation on your network. After starting the appropriate Server module, you supply the *QA*Load Script Development Workbench with the host name of the machine where the Server module is running and the port number that you specified when you started the Server module. When you are ready to run a test, start the Server module first. During the test NetLoad communicates with the NetLoad Server module, effectively loading the network. If NetLoad does not find the NetLoad Server module at the specified port—for instance if you mistyped the port number—the test fails (TCP) or fails to initiate (UDP).

Determining When to Use the TCP Server Module

If you are going to send TCP packets using NetLoad, you *must* have a *QA*Load TCP Server module running on each machine that you are sending packets to. Copy the TCP Server module file, NetloadTCPServer.exe, to each machine that will be receiving packets and double-click on the file to start the TCP Server module.

Because the *QA*Load TCP Server module is a Windows-based program, you cannot use it to send NetLoad TCP packets to a UNIX machine.

Determining When to Use the UDP Server Module

It is not necessary to have a *QA*Load UDP Server module running at the destination machine for NetLoad to successfully send packets to it; however, the Netload UDP Server can be useful to verify that the packets are being sent. To install the UDP Server module on a machine you are sending packets to, copy the program NetloadUDPServer.exe to that machine. Double-click the file to start the UDP Server module.

Since it is not necessary to have the UDP Server module running, you can send NetLoad UDP packets to both UNIX and Windows workstations.

Note

If you are testing UDP in "broadcast" mode, it is not necessary to use the NetLoad Server module.

Starting the NetLoad Server Module

If you are load testing a network running TCP/IP or UDP, the NetLoad Server module appropriate for your protocol should be running on a Windows 98 or later or NT workstation on your network *before* you start the test. The Server modules are installed with your *QA*Load product if you chose to install them during setup. If you are unsure if you should be using a NetLoad Server module, see "NetLoad Server Modules for TCP/IP and UDP" on page 5-2.

1. From the Start menu, choose **Programs>Compuware>QALoad>NetLoad**.
2. Select the appropriate Server module: TCP Server or UDP Server. The QALoad NetLoad Server dialog box opens.
3. In the **Specify Port Number** field, type the port number of the host machine. Click **OK**. The QALoad NetLoad Server window opens.
4. The QALoad NetLoad Server window will display messages indicating when the NetLoad Server receives packets. You can specify how often to receive messages from the Options menu by selecting one of the following:
   - **Show Message Every Packet** — Displays a message like the following upon receiving a packet: *Msg #980 8192 bytes*.
   - **Show Message Every 100 Packets** — Displays a message like the following every hundred packets: *100 Messages Received*.

Starting a NetLoad Session

Like all *QA*Load script development components, NetLoad is accessed from the *QA*Load Script Development Workbench.

1. From the *QA*Load Script Development Workbench, choose **Session>NetLoad**.
2. Open an existing protocol datapool file or create a new one:
   - To create a new datapool file, choose **File>New**. The New NetLoad File dialog box opens.
   - To open an existing datapool file, choose **File>Open**. The Open NetLoad File dialog box opens.
3. Select the protocol you wish to test on and click **OK**. If you are opening an existing datapool file, navigate to the file and open it.

   Note that the *QA*Load Script Development Workbench allows you to have multiple files open at the same time. Datapool files are located in the directory \Program Files\Compuware\QALoad\Middlewares\NetLoad\Scripts.

4. Enter or edit the appropriate datapool information in the Workbook Pane. For more information about NetLoad datapools, see "Creating a Datapool" on page 5-4.

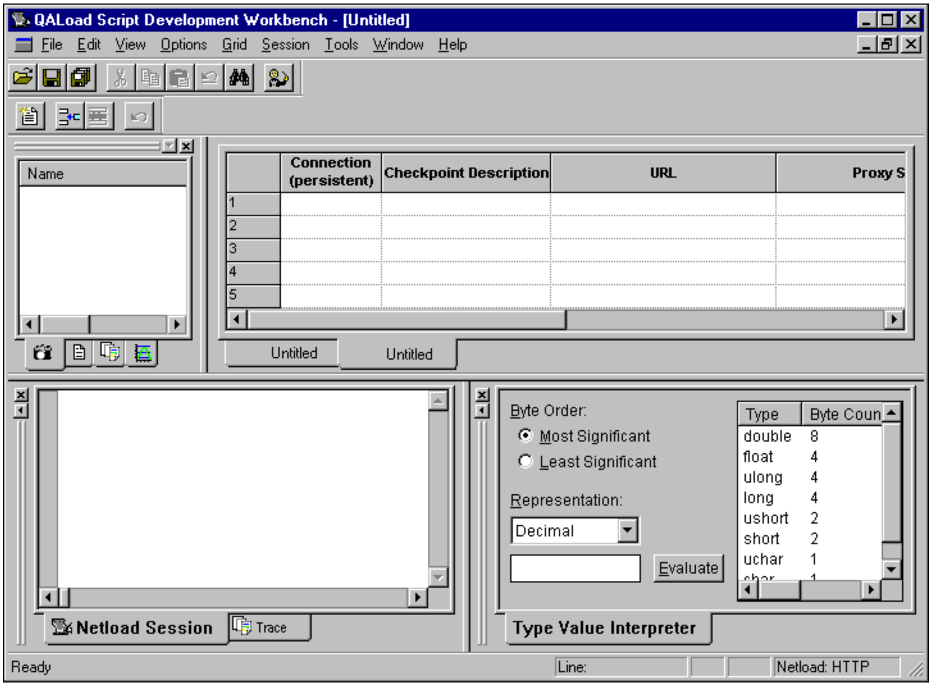
For more information about running a test, see Chapter 3, "Running a Load Test".

Creating a Datapool

You can change the behavior of the *QA*Load-provided script by creating datapool files (.dat) to insert into the script. For example, you can specify the URL to retrieve for HTTP testing or whether the script should abort when encountering an error. Using the *QA*Load Script Development Workbench, you can create customized datapool files for the protocol you will be testing. The *QA*Load Script Development Workbench provides a template for specific data to be entered for each datapool file.

Entering Datapool Data

1. From the *QA*Load Script Development Workbench, choose **Session>NetLoad**.
2. Select **File>New** to open the New NetLoad File dialog box. Select the protocol for which you wish to create a datapool file and click **OK**. A grid similar to the one shown below appears in the Workbook Pane. Each row on the grid represents a single data record. The column headings indicate the appropriate field information to enter. Note that the actual fields in the grid vary by protocol.



Byte Order:
● Most Significant
○ Least Significant

Representation: Decimal

Type	Byte Coun
double	8
float	4
ulong	4
long	4
ushort	2
short	2
uchar	1
char	1

3. Enter the appropriate information for your datapool file. Note that some fields on the grid contain pull-down menus. To activate them, click anywhere within the field. Then make your selection from the menu that appears.

4. When you are finished, select **File>Save** to name and save the datapool file. Note that your datapool file is listed in the Workspace Pane Datapools tab. *QA*Load creates a C-based script by the same name and lists it in the Workspace Pane Scripts tab. Both files will be saved to your \NetLoad\Scripts directory (for example, c:\Program Files\Compuware\QALoad\Middlewares\NetLoad \Scripts*datapool.dat*).

5. (Optional) Write a description of this datapool file for later reference by selecting **Options>NetLoad**. Once a description has been entered for a datapool file, you can review or edit the description any time the file is open by selecting **Options>NetLoad** again.

Datapool Fields

For a description of the datapool fields provided for your specific protocol, see the appropriate section below:

MSExchange

Checkpoint Description: A description of this checkpoint.

Profile Name: Type the name of your mail profile. For example, **Microsoft Outlook**.

Send To: Type the names of one or more mail recipients, separated by commas (,) or semi-colons (;).

Cc: Type the names of one or more mail recipients, separated by commas (,) or semi-colons (;).

Size of Body: Select a file size from the drop-down list for the body of the mail message.

Attached file size: Select a file size for the attachment file from the drop-down list.

FTP

Send/Receive: Specifies whether the script will be sending or receiving a file.

ASCII/Binary: Describes whether the file contains ASCII or binary data.

Checkpoint Description: A description of this checkpoint.

Host: The name of the host computer.

User ID: A user ID for accessing the host computer.

Password: A password for accessing the host computer.

File Size Options: Describes whether the file being sent to the host is of fixed or random size.

File Size (min): The minimum file size to send to the host or the size of the fixed file.

File Size (max): The maximum file size to send to the host.

Path: The path of the file to receive, or the destination of the file being sent. You must enter an absolute path.

Filename: The name of the file to receive or of the file being sent.

HTTP

Connection: Describes whether the connection is regular (the connection is closed after the request/response completes) or persistent (the connection remains open for subsequent requests).

Checkpoint Description: A description of this checkpoint.

URL: The address of the page to receive.

Proxy Server: The name of the proxy server (optional). Note that only proxy servers that *do not* require a user ID and password are supported.

PING

Checkpoint Description: A description of this checkpoint.

Host Name: The name of the host computer.

Pkt Size (Fixed/Random): Describes if the packet being sent to the host is of fixed or random size.

Pkt Size (min): The minimum packet size to send, or the size of the fixed packet to send.

Pkt Size (max): The maximum packet size to send.

LDAP

Checkpoint Description: A description of this checkpoint.

Host Name: The name of the host computer.

Search String: The text string to search for.

POP3

Checkpoint Description: A description of this checkpoint.

POP3 Server: The name of the POP3 server machine.

User ID: A user ID for accessing the POP3 server.

Password: A password for accessing the POP3 server.

Delete after read: Choose whether to delete the message after it has been read.

Connection: Describes whether the connection is regular (the connection is closed after the request/response completes) or persistent (the connection remains open for subsequent requests).

SMTP

Checkpoint Description: A description of this checkpoint.

SMTP Server: The name of the SMTP server machine.

From: Enter an email address or name.

Send To: Type the names of one or more mail recipients, separated by commas (,) or semi-colons (;).

Cc: Type the names of one or more mail recipients, separated by commas (,) or semi-colons (;).

Size of Body: Select a file size from the drop-down list for the body of the mail message.

File Path: Select a file from the drop-down list to use as the body of the mail message. This field displays files in the local directory only if you selected Browse in the Size of Body field.

Attached file size: Select a file size for the attachment file from the drop-down list.

Attached file path: Select a file from the drop-down list to use as an attachment. This field displays files in the local directory only if you selected Browse in the Attached File Size field.

Connection: Describes whether the connection is regular (the connection is closed after the request/response completes) or persistent (the connection remains open for subsequent requests).

TCP

Checkpoint Description: A description of this checkpoint.

Host Name: The name of the host computer.

Port: The port number of the host computer.

Pkt Size (Fixed/Random): Describes if the packet being sent to the host is of fixed or random size.

Pkt Size (min): The minimum packet size to send, or the size of the fixed packet to send.

Pkt Size (max): The maximum packet size to send.

UDP

Checkpoint Description: A description of this checkpoint.

Host Name: Type the name of the host computer that is to receive the packet.

Port: The port number of the host computer.

Pkt Size (Fixed/Random): Describes if the packet being sent to the host is of fixed or random size.

Pkt Size (min): The minimum packet size to send, or the size of the fixed packet to send.

Pkt Size (max): The maximum packet size to send.

Editing a Datapool

1. With the appropriate NetLoad session open, open the script by choosing **File>Open** and navigating to it, or select it from the Workspace tab Datapools tab.
2. Make any changes or additions to the file.
    - To delete an entire record (a single row), click its row number and select **Grid>Delete Row(s)** or press the Delete key on your keyboard.
    - To insert a new record (a single row) above an existing record, click a row number and select **Grid>Insert Row** or press the Insert key on your keyboard. NetLoad inserts a blank row above the selected row.
3. Save any changes to the file by selecting **File>Save**.

For information about running a test, see Chapter 3, “Running a Load Test”.

Verifying CDO Support for MSExchange

Before you can successfully test with NetLoad for MSExchange using Outlook 2000, you must ensure that Collaboration Data Objects (CDO) support is installed.

1. From the Windows task bar, select **Start>Settings>Control Panel**.
2. Double-click the **Add/Remove Programs** icon.
3. From the list on the Install/Uninstall tab, select **Microsoft Office 2000** or **Microsoft Outlook 2000**.
4. Click the **Add/Remove** button.
5. Click **Add or Remove Features**.
6. Click the plus sign (+) next to Microsoft Outlook for Windows.
7. Select **Collaboration Data Objects**, and then click **Run from My Computer**.

Chapter 6. Analyzing Load Test Results

This chapter provides information for analyzing load test results using *QA*Load Analyze It includes the following sections:

- **Accessing Test Results —** Describes *QA*Load timing files and how to access them.
- **Viewing Test Data —** Describes the files and reports available to view test data results.
- **Displaying Detail Data —** Describes how to create a detail view to display detailed timing data from timing files.
- **Graphing Test Results —** Describes how to use *QA*Load Analyze's charting features to display checkpoints graphically.
- **Integrating ServerVantage Agent Data —** Describes how to access and graph ServerVantage Agent data collected during a load test.
- **Viewing Application Expert and QALoad Integrated Reports —** Describes how to view Application Expert and *QA*Load integrated reports.
- **Exporting Test Results to CSV or HTML Formats or RIP files —** Describes how to export selected test data to comma separated value (CSV) or HTML format or export RIP files for viewing and reporting purposes.
- **Emailing Test Results —** Describes how to email test results saved as HTML or *QA*Load Analyze timing files using Microsoft Exchange.

Accessing Test Results

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 merges that data into the timing file, as well.

Primary timing files are saved in the directory **\Program Files\Compuware\QALoad\TimingFiles**, and use the following naming convention:

- the *most recent* test run will always take the name *sessionID*.tim, where *sessionID* is the name of the session ID used in the test run.
- subsequent test runs will be named *sessionIDxxx*.tim, where *xxx* is a number representing the order in which the tests were run.

In other words, the first of multiple test runs will *always* be named sessionID001.tim, and the last (most recent) of multiple test runs will always be named sessionID.tim.

The Primary timing file created by the Conductor after a test run contains all of the timing records of all Players in that test run. Use *QA*Load Analyze to view, sort, graph, and create reports using the test data in the timing file.

The following pages explain how to open timing files and view and sort *QA*Load test data.

Hint

In the event that something goes wrong on the network and a Player timing file is not passed to the Conductor, it is still possible to analyze results from a Player timing file. Player timing files are saved in the directory **\Program Files\Compuware\QALoad\TimingFiles** and are named tim_*yyyymmdd_hhmmss_xxx*.ptf, where *yyyymmdd_hhmmss* is the date/time the test was started, and *xxx* is the Player number.

Opening Test Results

Note

*QA*Load Analyze results look best when you use a color palette greater than 256 colors on your desktop.

*QA*Load Analyze offers three ways to access and open a timing file containing test results:

***QA*Load Conductor**

— From the **QALoad Conductor's** menu, select **Tools>Options**. The Options dialog box appears.
— Click the **General** tab. In the **General Options** area, select the **Launch Analyze After Test** check box.

At the end of each test run, *QA*Load Conductor will automatically launch *QA*Load Analyze and open the most recent timing file. Or, if you did not select the **Launch Analyze After Test** check box before the test:

- Select **Tools>Analyze**.
- From the ***QA*****Load Analyze** menu, select **File**>**Open**. The Open timing file dialog box appears. Select a timing file to work with by double-clicking the filename in the list of available timing files.

*QA*Load Conductor automatically passes the current test's timing file to *QA*Load Analyze. Start *QA*Load Analyze from *QA*Load Conductor at the end of a load test.

Windows Start menu

- Click the taskbar's **Start** button and choose **Programs>Compuware>QALoad>Analyze**.
- From the ***QA*****Load Analyze** menu, select **File>Open**. The Open timing file dialog box appears. Select a timing file to work with by double-clicking the filename in the list of available timing files.

Use this method when you need to access a previously created timing file.

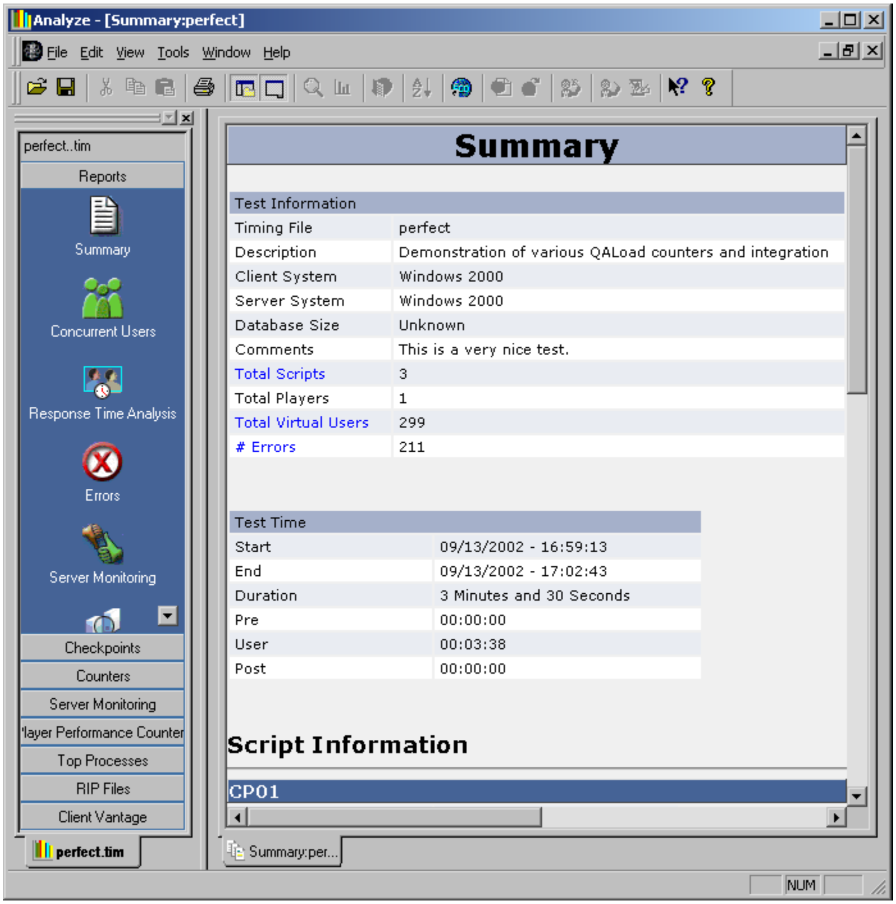
QA**Load Script Development Workbench**

- From the ***QA*****Load Script Development Workbench** menu, select **Tools>Analyze**.
- From the ***QA*****Load Analyze** menu, select **File**>**Open**. The Open timing file dialog box appears. Select a timing file to work with by double-clicking the filename in the list of available timing files.

Use this method when you are already working in the *QA*Load Script Development Workbench and need to access a previously created timing file.

Displaying Test Data in *QA*Load Analyze

When you open a timing file, *QA*Load Analyze displays test data in its interactive split-window view. Use *QA*Load Analyze's interactive view to sort test data, view detailed checkpoint data, and produce a variety of graphs and reports. Each window is described in the following sections.



Summary

Test Information	
Timing File	perfect
Description	Demonstration of various QALoad counters and integration
Client System	Windows 2000
Server System	Windows 2000
Database Size	Unknown
Comments	This is a very nice test.
Total Scripts	3
Total Players	1
Total Virtual Users	299
# Errors	211

Test Time	
Start	09/13/2002 - 16:59:13
End	09/13/2002 - 17:02:43
Duration	3 Minutes and 30 Seconds
Pre	00:00:00
User	00:03:38
Post	00:00:00

Script Information

CP01

Figure 6-1. *QA*Load Analyze Split-Window View

Workspace

The Workspace is the starting point for generating reports and graphs available from *QA*Load Analyze. It is a dockable window containing timing file tabs with groups corresponding to the active timing file. Each group displays data from a timing file, most in hierarchical tree-views for a clear, logical test data representation. The following groups may be available:

- Reports
- Checkpoints
- Counters
- Server Monitoring
- Player Performance Counters
- Top Processes

- RIP Files

Note

The number and type of groups available in the Workspace is dependent on the data collected in the timing file (which is determined by the *QA*Load Conductor option selected at the time of running a Load Test).

Use the Workspace to select data from which to generate reports, display detail views, or create graphs. From the Workspace tree-view, you can:

- display details for single or multiple data points (see “Displaying Detail Data” on page 6-10).
- create a graph plotting data point detail (see “Graphing Test Results” on page 6-13).
- access and view ServerVantage Agent data (see “Displaying ServerVantage Agent Data” on page 6-26).
- send an email message with pre-defined reports and associated files or a timing file attached (see “Emailing Test Results” on page 6-31).

The sections that follow provide information about options for displaying, sorting, graphing, and creating reports from load test timing data.

Hint

Since *QA*Load Analyze is a Multiple Document Interface (MDI), you can open multiple timing files at the same time.

Data Window

The Data window dynamically displays pre-defined reports, detail views, and graphs of selected test data. When you open a timing file, the Data window automatically displays a Summary report that includes test information from the session ID file and timing information from all checkpoints. For a detail view of a load test, select a group, such as Checkpoints and select the desired checkpoints from the tree-view, then select **View>Detail**. To graph results of a load test, from the **View** menu, choose **Graph**. For information on graphing timing data, see “Graphing Test Results” on page 6-13.

Viewing Test Data

An important part of the load testing process is viewing and studying the results of a test. View load test results not only on a machine where *QA*Load is installed, but also on any machine with a Web browser.

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

The following files are found in the working folder:

Table 6-1. Working Folder Files

File Name	Description
<timingfile>.xml.source	Working folder generated in the reports folder when opening a timing file. The working folder name is always the <name of the timing file> with a .xml.source extension.
<timingfile>.xml	Original timing file with just enough information to create the *QA*Load Analyze pre-defined reports. It is a representation of the timing file, <timingfile>.tim.
<timingfile>.complete.xml	Original timing file containing all data collected during a load test. It can be an extremely large file. Use this file if creating a report using XSL that required this data.
summary.htm	Use this HTM file to view the Summary report, or any other available pre-defined report, in any Web browser.
summary.xml	Generated XML file for the Summary report, or any other available pre-defined report.
summary.xsl	Generated XSL file for the Summary report, or any other available pre-defined report. Translates the .xml file specifying HTML as its output and generates the HTML report. Use this file to customize the reports by writing in .xsl.
default.htm	Report which provides a main screen to launch any other pre-defined reports. Uses nav.htm for the navigation frame.

Custom Reports

*QA*Load Analyze provides the ability to create custom reports using XML (Extensible Markup Language), XSL (Extensible Style Language), and HTM (Hypertext Markup) files. As seen in Table 6-1 on page 6-6, *QA*Load Analyze provides a set of files in .htm, .xml, and .xsl formats in addition to the .tim file. *QA*Load Analyze automatically generates a XML (*.xml), XSL (*.xsl), HTM (*.htm), file when you open a timing file.

Pre-defined Reports

*QA*Load 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 available depend on the data collected in the timing file (which is determined by the *QA*Load Conductor option you select at the time of running the Load Test.)

The following pre-defined reports may be available:

- **Summary**— Provides basic summary information from a load test. Presents timing information for each transaction in the timing file and the minimum, maximum, and median response times for each checkpoint.
- **Session**— Provides summary information about the test session. Presents test statistics, script settings, and checkpoint statistics.
- **Concurrent Users**— Displays a graph of the total virtual users (concurrent users vs. elapsed time) for the test and for individual scripts that were part of the test.
- **Response Time Analysis**— Displays of graph of each script's transaction duration (response time vs. elapsed time) as well as checkpoint summary data.
- **Errors**— Provides a cumulative list of all errors, sorted by script and occurrence in time, that occurred during the course of the load test.
- **Client Throughput**— Provides a graph of HTTP Reply analysis for key HTTP counters - HTTP counter vs. elapsed time.
- **Server Monitoring**— Provides Remote Monitoring, Server Analysis Agent, and ServerVantage performance counter information.
- **Transaction Throughput**— Provides the cumulative number of transactions over elapsed time for each script and for the total test. Also provides the incremental number of transactions per specified period over elapsed time.

Understanding Durations

When you begin to analyze your test results, it is important to understand how durations are calculated by *QA*Load.

Transaction duration

Transaction duration is the time that the server under test takes to complete a transaction, from the `BEGIN_TRANSACTION` command to the `END_TRANSACTION` command.

Three factors comprise transaction duration:

- The script processing time including, but not limited to, added script logic, *QA*Load processing of server replies, and other *QA*Load processing.
- Sleep time, if the Conductor's **Do not include Sleep times in Checkpoint Timings** option is not selected.
- The response time of the application under test, including, but not limited to, the application server, database access, and network.

Checkpoint duration

Checkpoint duration is the amount of time between begin and end checkpoint statements.

If you select the Conductor's **Enable Automatic Middleware Checkpoint Timings** option or use the `BeginCheckpoint` and `EndCheckpoint` functions in the script, the following factors comprise checkpoint duration:

- The response time of the application under test, including, but not limited to, the application server, database access, and network.
- Sleep time, if the Conductor's **Do not include Sleep times in Checkpoint Timings** option is not selected.
- *QA*Load processing time is not included within these checkpoints.

If you use the `BEGIN_CHECKPOINT` and `END_CHECKPOINT` methods in the script, the checkpoint duration is calculated by the actual clock time from when `END_CHECKPOINT` was issued minus the actual clock time from `BEGIN_CHECKPOINT`.

Note

`BEGIN_CHECKPOINT` is a deprecated command. Compuware recommends using the `BeginCheckpoint` function.

The sum of all checkpoint durations may not match the value of the transaction duration in the timing file. The following examples illustrate situations in which this may occur:

If the script includes nested checkpoints, the sum of the checkpoint durations is greater than the transaction duration. For example:

```
BEGIN_TRANSACTION();
BeginCheckpoint(“Checkpoint1”);
BeginCheckpoint(“Checkpoint2”);
DO_SLEEP(5);
EndCheckpoint(“Checkpoint2”);
EndCheckpoint("Checkpoint1");
END_TRANSACTION();
```

In this example, Checkpoint1 and Checkpoint2 are timing the same items. The sum is greater than the transaction duration because the Conductor's **Do not include Sleep times in Checkpoint Timings** option is not selected, and the sleep factor is set to a value of greater than 0%.

If the Conductor's **Enable Automatic Middleware Checkpoint Timings** option is selected and you have created your own checkpoint statements in the script, the sum of the checkpoint durations is greater than the transaction duration. For example:

```
BEGIN_TRANSACTION();
BeginCheckpoint("checkpoint1");
DO_Http("");
EndCheckpoint("checkpoint1");
END_TRANSACTION();
```

In this example, a checkpoint pair surrounds a `DO_Http` statement. If the Conductor's **Enable Automatic Middleware Checkpoint Timings** option is selected, there will be two checkpoint statements in the timing file for the same items.

If you insert a checkpoint pair before or after the transaction loop in the script, the sum of the checkpoint durations is greater than the transaction duration. For example:

```
BeginCheckpoint("Checkpoint1");
EndCheckpoint("Checkpoint1");
BEGIN_TRANSACTION();
END_TRANSACTION();
BeginCheckpoint("Checkpoint2");
EndCheckpoint("Checkpoint2");
```

If you add scripting logic to the script, the sum of the checkpoint durations can be less than the transaction duration. For example:

```
BEGIN_TRANSACTION();
BeginCheckpoint("Checkpoint1");
Response = DO_Http();
EndCheckpoint("Checkpoint1")
// Do custom processing of response (about 5 seconds)
ProcessTheResponse(Response);
END_TRANSACTION();
```

In this example, a user-defined command named `ProcessTheResponse` has been added to the script. This command processes a response from a server reply, which could take an additional five seconds.

If the amount of *QA*Load processing time is high, the sum of the checkpoint durations can be less than the transaction duration. For example:

```
BEGIN_TRANSACTION();
BeginCheckpoint("Checkpoint1");
```

```
DO_Http();
EndCheckpoint(“Checkpoint1”)
END_TRANSACTION();
```

In this example, assuming the server response is complex, *QA*Load’s processing time would increase. This increased processing time would not be included in the checkpoint duration, but it would be included in the transaction duration.

Displaying Detail Data

Using the Workspace as a starting point, display detailed timing data from a timing file such as checkpoints, counters, etc. in the Data window. This view, containing both a summary and data table, lets you view statistics not only for the active timing file, but also for other timing files. *QA*Load Analyze allows you to drag and drop additional items from any timing file group, such as checkpoints or counters, onto the active view. Combine data from different groups for the same timing file or combine data from different timing files.

Creating a Detail View

To display this view, with the appropriate timing file tab selected:

1. In the Workspace, click the **group** for which you wish to view statistics.
2. Select data points (checkpoints, counters, etc.) by clicking the selection check box for each data point. To select all data points, choose **Edit>Select all** from the ***QA*Load Analyze** menu.
3. From the **View** menu, select **Detail** or click the **View Detail** toolbar button.

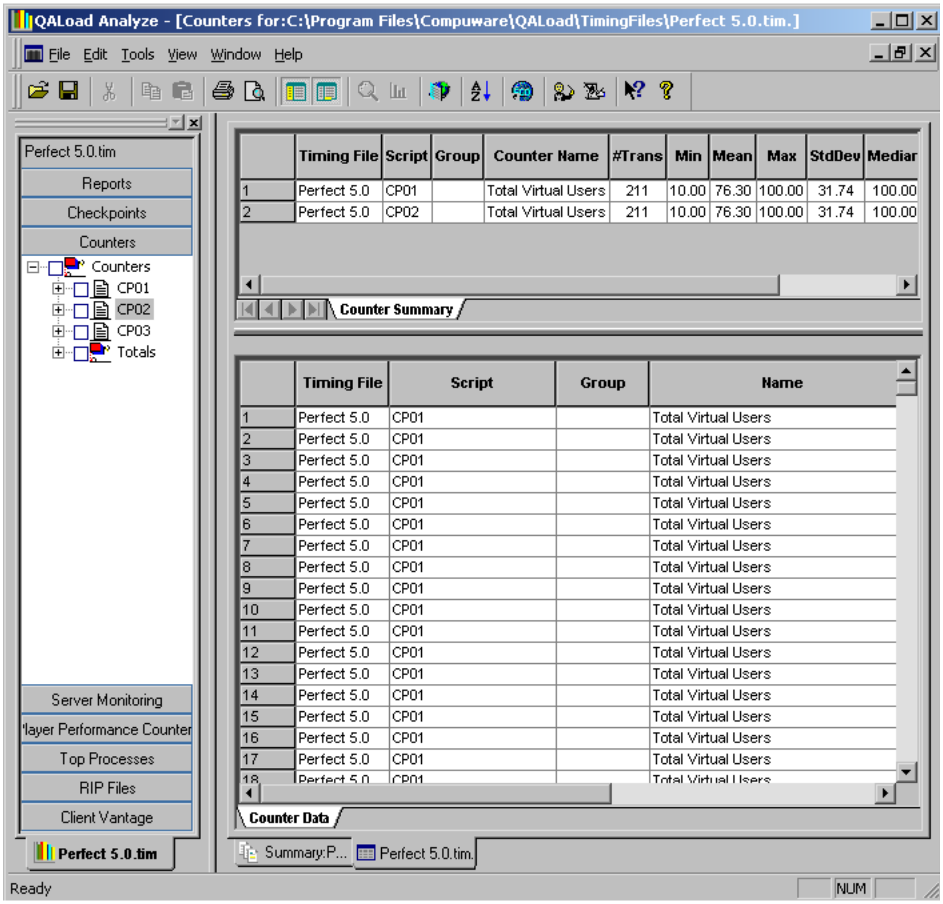
Detailed information appears in the Data window in both a Summary and Details table. The information displayed will vary depending on the group selected. A tab also appears in the bottom of the Data window indicating the active timing file. The following detail views are available:

- Checkpoints
- Counters
- Server Monitoring:
  - EcoTOOLS6
  - Remote Monitoring data
  - Server Analysis data
  - ServerVantage data

- Player Performance Counters
- Top Processes data

While the information displayed will vary depending on the group selected, all detail views share the following statistical details:

- **#Trans** — Total number of data points that were used to calculate the statistics.
- **Min** — Minimum recorded response time.
- **Mean** — Average of the response times.
- **Max** — Maximum-recorded response time.
- **StdDev** — Standard deviation of all response times. Standard deviation is an indicator of how widely values are dispersed from the average (mean) value. A large standard deviation indicates a wide variance in response times.
- **Median** — Response time (in seconds). The median is the value at which half of the responses are greater and half are less. If the number of responses is large, the median is usually close to the mean.
- **Nth Percentile** — Displays that *nth*% of the responses have a value less than the value shown.


QALoad Analyze - [Counters for:C:\Program Files\Compuware\QALoad\TimingFiles\Perfect 5.0.tim.]

	Timing File	Script	Group	Counter Name	#Trans	Min	Mean	Max	StdDev	Mediar
1	Perfect 5.0	CP01		Total Virtual Users	211	10.00	76.30	100.00	31.74	100.00
2	Perfect 5.0	CP02		Total Virtual Users	211	10.00	76.30	100.00	31.74	100.00

	Timing File	Script	Group	Name
1	Perfect 5.0	CP01		Total Virtual Users
2	Perfect 5.0	CP01		Total Virtual Users
3	Perfect 5.0	CP01		Total Virtual Users
4	Perfect 5.0	CP01		Total Virtual Users
5	Perfect 5.0	CP01		Total Virtual Users
6	Perfect 5.0	CP01		Total Virtual Users
7	Perfect 5.0	CP01		Total Virtual Users
8	Perfect 5.0	CP01		Total Virtual Users
9	Perfect 5.0	CP01		Total Virtual Users
10	Perfect 5.0	CP01		Total Virtual Users
11	Perfect 5.0	CP01		Total Virtual Users
12	Perfect 5.0	CP01		Total Virtual Users
13	Perfect 5.0	CP01		Total Virtual Users
14	Perfect 5.0	CP01		Total Virtual Users
15	Perfect 5.0	CP01		Total Virtual Users
16	Perfect 5.0	CP01		Total Virtual Users
17	Perfect 5.0	CP01		Total Virtual Users
18	Perfect 5.0	CP01		Total Virtual Users

Figure 6-2. Detail View

Sorting Information on the Detail View

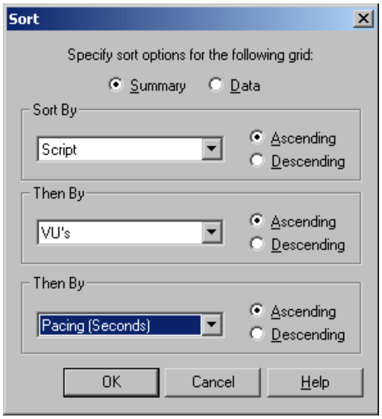
A Detail view potentially contains a large number of checkpoints, counters, etc., especially if a load test had many virtual users. To make the information manageable, specify up to three levels of criteria to sort by, in ascending or descending order.

For example, if a test ran using five scripts on one-hundred virtual users, sort the data by script name. Suppose each virtual user ran more than one transaction using a particular script, then sort by both script name and by virtual user. Or, to quickly locate any timing bottlenecks, sort by response time.

Sort checkpoint data by up to three criteria as follows:

1. Click anywhere in the Detail view.

2. Click the **Sort** button or choose **Tools>Sort** from the menu to open the Sort Grid dialog box.



Sort

Specify sort options for the following grid:

● Summary ○ Data

Sort By: Script ● Ascending ○ Descending

Then By: VU's ● Ascending ○ Descending

Then By: Pacing (Seconds) ● Ascending ○ Descending

3. Select the first-, second-, and third-level sort criteria from the **Sort By**, and **Then By**, drop-down lists, selecting the appropriate sort order — **Ascending** or **Descending —** for each.

Note

Select *none* at any level if you do not wish to specify a sort order.

4. Click **OK**. The data in the Detail view is sorted according to the criteria specified.

Graphing Test Results

With the Workspace as a starting point, use *QA*Load Analyze's charting features to graph timing data in a number of formats and styles. The following sections describe how to determine the amount of data to graph, how to create a graph, and how to customize a graph.

Managing Large Amounts of Data

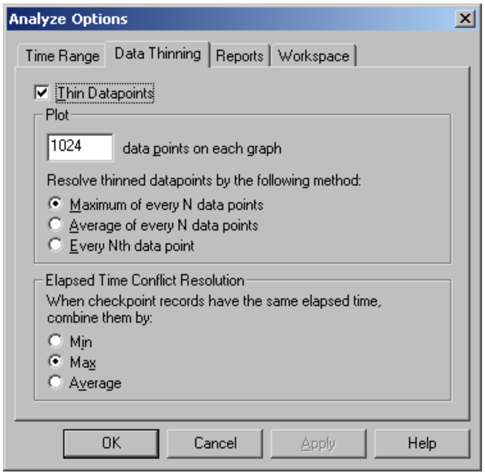
With a large number of virtual users, it's possible to create a timing file containing hundreds of thousands of timing records for each checkpoint. Attempting to graph just a few of those checkpoints can slow *QA*Load Analyze down considerably. For example, if a timing file contained 250,000 timing records for each data point, attempting to graph even one checkpoint means that *QA*Load 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, *QA*Load 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, *QA*Load Analyze provides an option that allows you to determine how to *thin* data. That is, how to determine how many data points to plot.

Set data thinning options before attempting to graph large amounts of data:

1. With a timing file open, select **Tools**>**Options**. The Analyze Options dialog box opens.



Analyze Options

☑ Thin Datapoints

Plot

1024 data points on each graph

Resolve thinned datapoints by the following method:

● Maximum of every N data points
○ Average of every N data points
○ Every Nth data point

Elapsed Time Conflict Resolution

When checkpoint records have the same elapsed time, combine them by:

○ Min
● Max
○ Average

2. Click the **Data Thinning** tab. Enter the data points to plot on each graph. This value should typically be no more than the resolution of the screen width. However, you may want to enter a smaller number to limit the number of data points that appear on the graph.

3. Select a method of plotting based on the number entered in Step 2:

   - **Maximum of every N data points** — Plots the maximum value of every *N* points. *N* is calculated as the total number of transactions divided by the number of points that can be plotted. This method insures that any peaks in the data will be plotted.

   - **Average of every N data points** — Plots the average of every *N* points. This method gives a good approximation of the data. Any spikes in the data are averaged into the surrounding points.

- **Every Nth data point** — Plots every *N*th data point on the graph without any additional calculations. This method is slightly faster, but may hide changes in the response times (spikes, dips, etc.).

4. When finished, click **OK** to save the selections. The data thinning method specified becomes the default for all graphs until you change it again.

Creating a Graph

If you haven't considered thinning the amount of data to be shown on a graph, refer to "Managing Large Amounts of Data" on page 6-13 before using the following procedure to graph checkpoint data.

1. In the Workspace, select the checkpoint(s) to graph. If you intend to plot a response time distribution graph or a cumulative response time distribution graph, select a single checkpoint only.

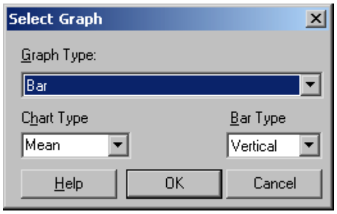
2. Click the **View Graph** button or choose **Graph** from the **View** menu to open the Select Graph Type dialog box.

3. Select a graph type. For a detailed description of each graph type, see "QALoad Analyze Graph Types" on page 6-18.

4. Click **OK**. You may need to take additional steps, depending on the type of graph you chose to create:

   - If you selected **Line**, data for the selected checkpoints are presented in a line graph format in *QA*Load Analyze's Data window. No additional steps are necessary. Instead, see the section "Where to Go Next" on page 6-18.

   - If you selected **Bar**, the Select Graph dialog box expands:



Select Graph

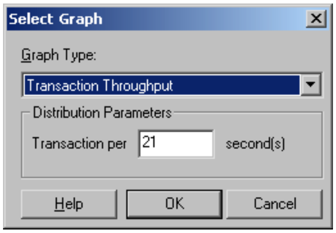
Graph Type: Bar

Chart Type: Mean

Bar Type: Vertical

— In the **Chart Type** area, select which criterion to base the data on: mean, median, or percentile response times. If you choose **Percentile Response Time**, type a value in the **Percentile** field.
— Select whether the graph should be horizontally or vertically oriented.
— Click **OK**. Data for the selected checkpoints are presented in a bar chart format in *QA*Load Analyze's Data window. Read the section "Where to Go Next" on page 6-18.

- If you selected **Transaction Throughput**, the Select Graph dialog box expands.



Select Graph

Graph Type: Transaction Throughput

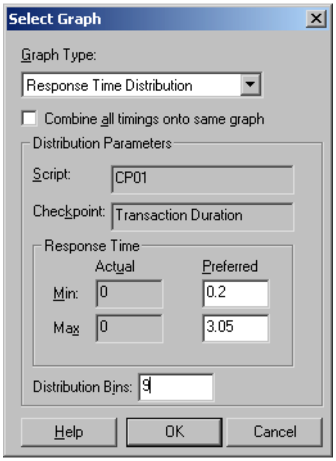
Distribution Parameters

Transaction per 21 second(s)

— In the **Distribution Parameters** area, type the number of seconds in the **Transaction per** <xx> **seconds** field.

— Click **OK**. Data for the selected checkpoints are presented in a line graph format in *QA*Load Analyze's Data window. Read the section "Where to Go Next" on page 6-18.

- If you selected **Response Time Distribution**, the Select Graph dialog box expands.



Select Graph

Graph Type: Response Time Distribution

□ Combine all timings onto same graph

Distribution Parameters

Script: CP01

Checkpoint: Transaction Duration

Response Time

	Actual	Preferred
Min:	0	0.2
Max	0	3.05

Distribution Bins: 9

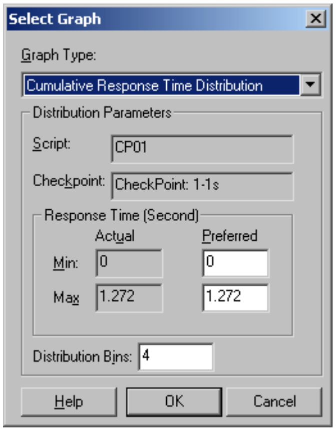
— In the **Response Time (Second)** area, enter a minimum response time to graph, if you wish. By default, this is set to the minimum time found within the dataset. Changing this value allows you to plot the distribution within a particular range.

— In the **Response Time (Second)** area, enter a maximum response time to graph, if you wish. By default, this is set to the maximum time found within the dataset. Changing this value allows you to plot the distribution within a particular range.

— In the **Distribution Bins** field, specify how many points to display in the graph. Each point will show the percentage of transactions with a response time that falls within a specific time range.

— Click **OK**. Data for the selected checkpoints are presented in a response time distribution graph, according to the parameters specified, in *QA*Load Analyze's Data window. Read the section "Where to Go Next" on page 6-18.

- If you selected **Cumulative Response Time Distribution**, the Select Graph dialog box opens.



Select Graph

Graph Type: Cumulative Response Time Distribution

Distribution Parameters

Script: CP01

Checkpoint: CheckPoint: 1-1s

Response Time (Second)

	Actual	Preferred
Min:	0	0
Max	1.272	1.272

Distribution Bins: 4

— In the **Response Time (Second)** area, enter a minimum response time to graph, if you wish. By default, this is set to the minimum time found within the dataset. Changing this value allows you to plot the distribution within a particular range.

— In the **Response Time (Second)** area, enter a maximum response time to graph, if you wish. By default, this is set to the maximum time found within the dataset. Changing this value allows you to plot the distribution within a particular range.

— In the **Distribution Bins** fields, specify how many points to display in the graph. Each point will show the percentage of transactions with a response time that falls within a specific time range.

— Click **OK**. Data for the selected checkpoints are presented in a cumulative response time distribution graph, according to the parameters specified, in *QA*Load Analyze's Data window. Read the section "Where to Go Next" on page 6-18.

Hint

To improve the visibility of any graph, choose **View>Workspace** or **View>Workbook** from the *QA*Load Analyze menu to temporarily hide the Workspace or timing file tabs. The Data window will expand into the additional space, enlarging the graph.

Where to Go Next

- Use *QA*Load Analyze's many features to customize a graph, see "Customizing Graphs" on page 6-23.
- To convert a graph to comma-separated value (CSV) or HTML format for viewing or reporting purposes, see "Exporting Test Results to HTML" on page 6-30.

*QA*Load Analyze Graph Types

The following basic graph types are available from *QA*Load Analyze. After generating one of the following graph types, you may further customize a graph in a number of ways. For details, see "Customizing Graphs" on page 6-23.

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.

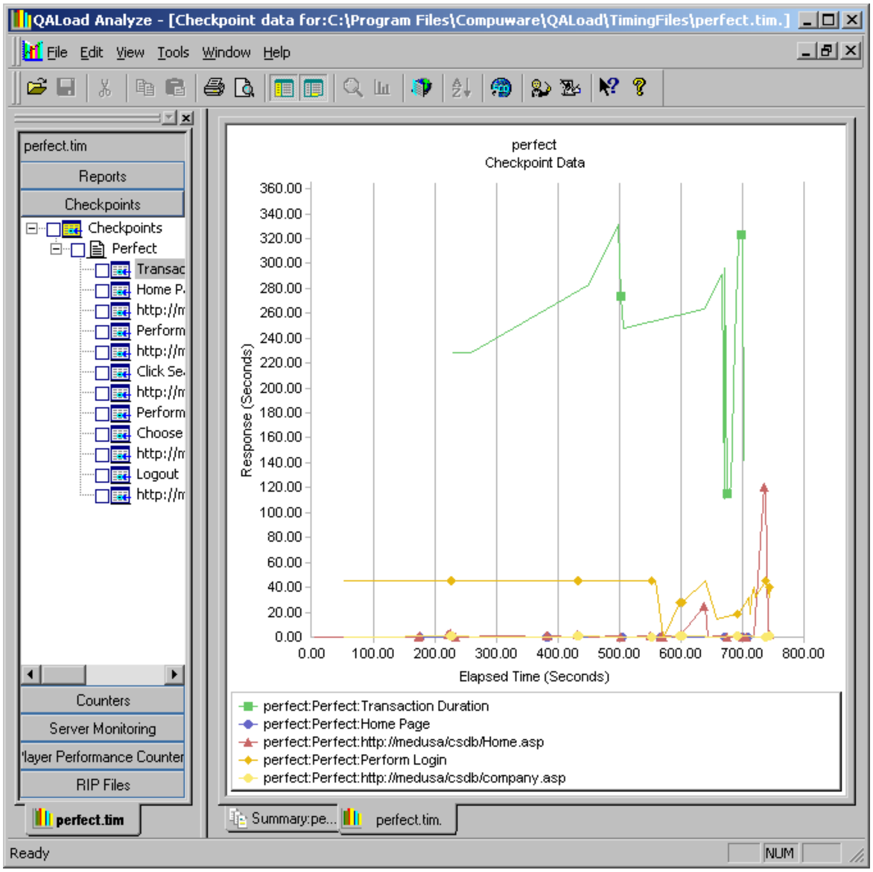


Figure 6-3. Line Graph

Bar Graph

A bar graph shows the median, mean, or percentile response times for the selected checkpoints. Figure 6-4 shows a bar graph displaying median response times.

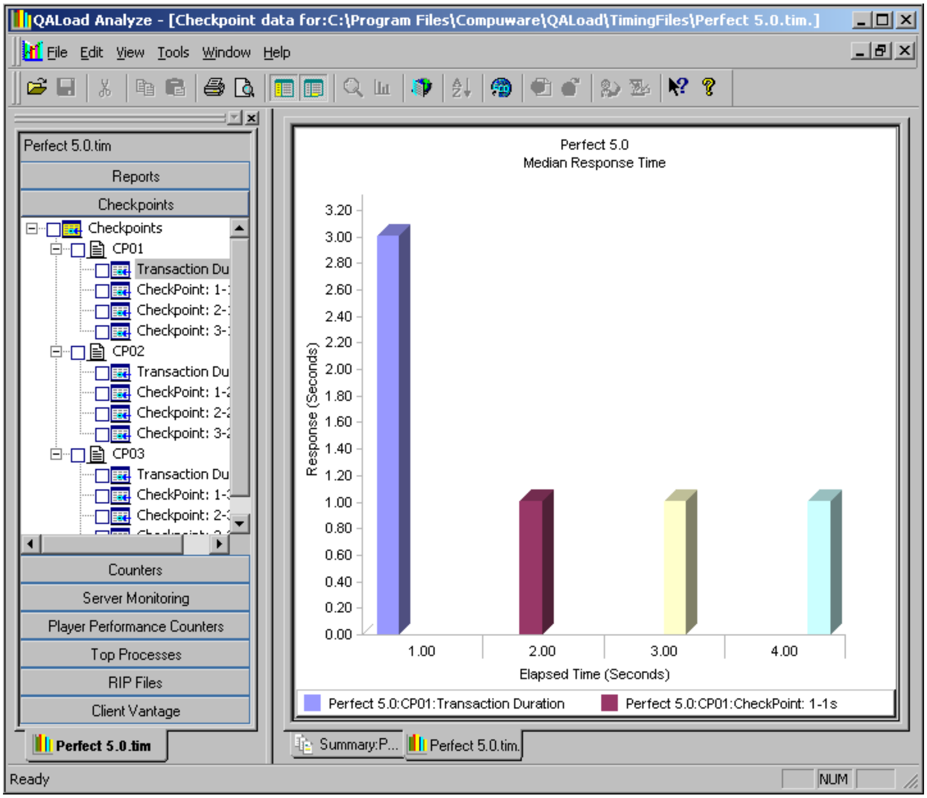


Figure 6-4. Bar Graph

Transaction Throughput Graph

Figure 6-5 shows a 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.

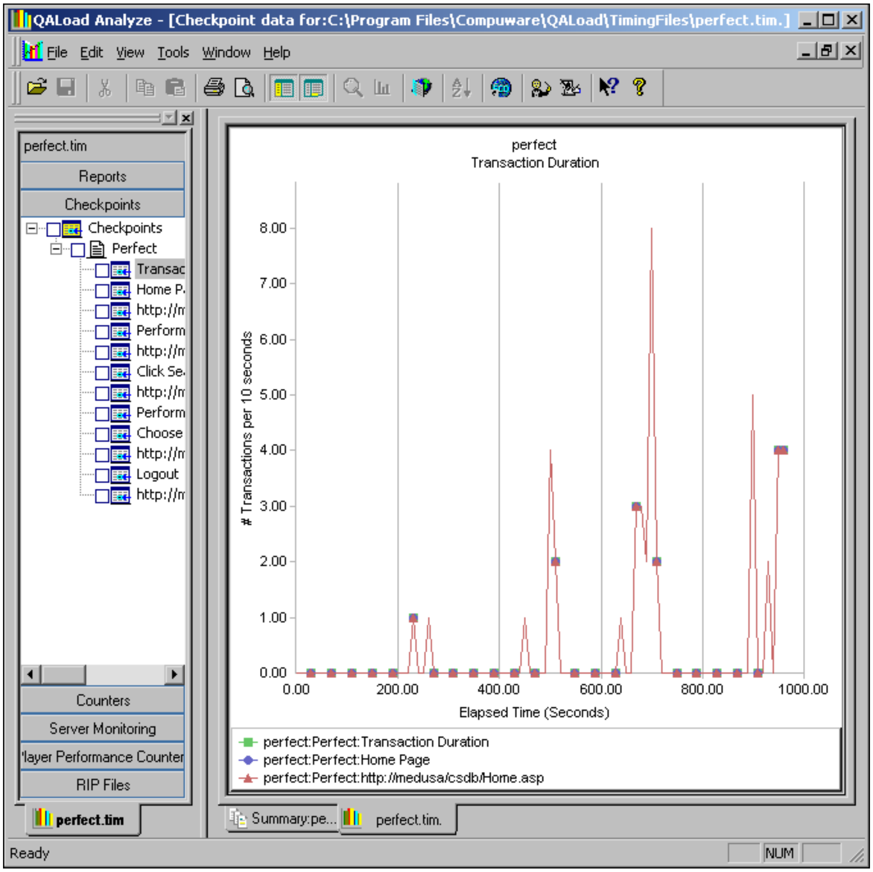


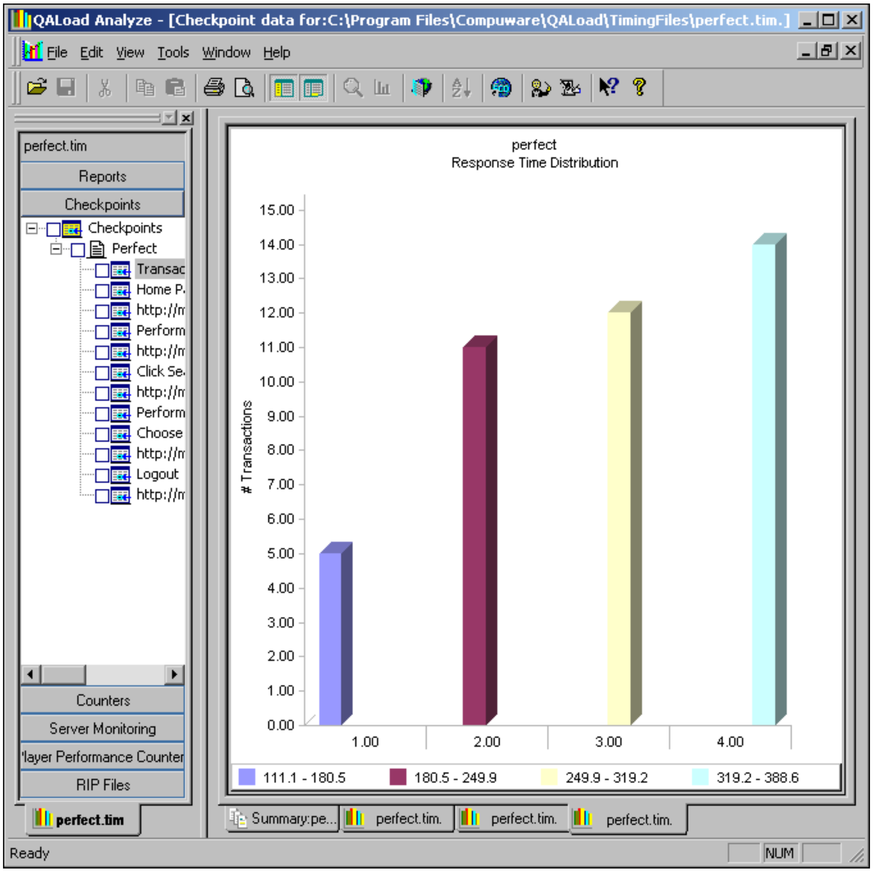
Figure 6-5. Transaction Throughput Graph

Response Time Distribution Graph

Figure 6-6 shows a 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.

Note

A response time distribution graph only shows results for a single checkpoint, although it can compare results from multiple timing files.



QALoad Analyze - [Checkpoint data for:C:\Program Files\Compuware\QALoad\TimingFiles\perfect.tim.]

Figure 6-6. Response Time Distribution Graph

Cumulative Response Time Distribution Graph

Figure 6-7 shows a 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.

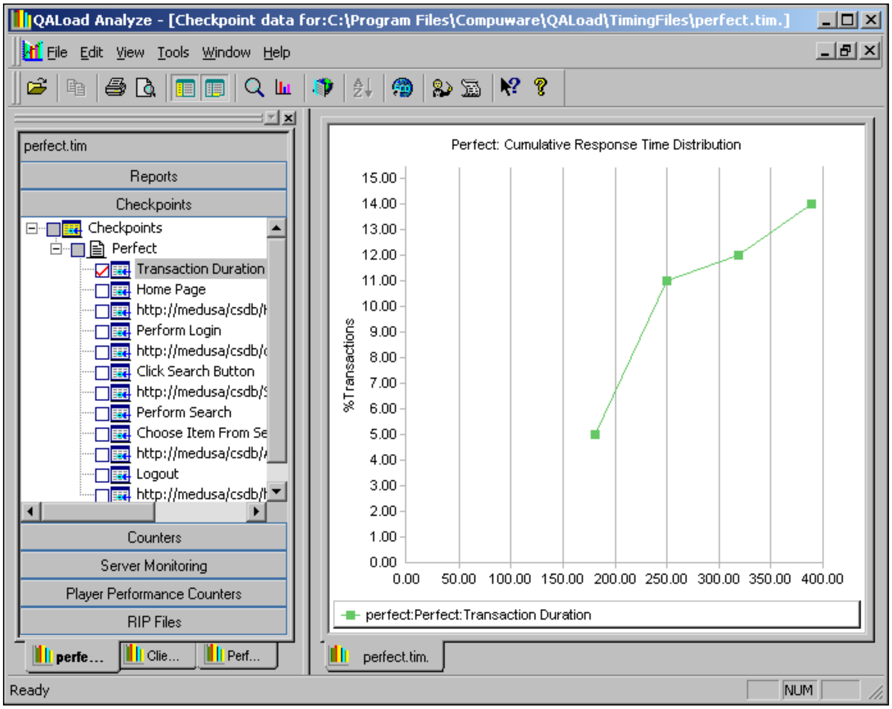


Figure 6-7. Cumulative Response Time Distribution Graph

Customizing Graphs

Change the style and appearance of a graph using options available from the Graph toolbar, shown in Figure 6-8. The Graph toolbar contains buttons for standard Windows operations as well as for customizing the appearance of your graphs. Although it normally appears atop of a graph, the toolbar is completely dockable. Move the toolbar to another side of the graph, or off the graph altogether, by clicking any unpopulated area of the toolbar and dragging it to another area.

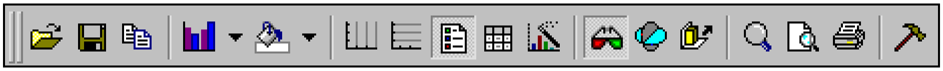


Figure 6-8. QALoad Analyze Graph Toolbar

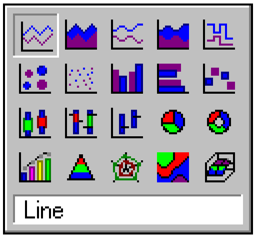
Customize a graph with the following features from the Graph toolbar:

- Graph type (gallery type)
- Color
- Grid orientation (horizontal and vertical)
- Legend box
- Data display

- Dimension (3D or 2D)
- Rotation
- Z-Cluster
- Color/pattern

Graph Type

To change the way the data is displayed in a graph, click the **Gallery** button on the graph toolbar. A graph type icon box (see Figure 6-9) displays. Select the desired graph type.



Line

Figure 6-9. Graph Type Icon Box for a Line Graph

The following graph types are available. The order and number of options available depends on which type of graph you created. For example, if you created a bar graph the only options available are Bar and Gantt.

Line	Area	Curve	Area-curve	Step
Bubble	Scatter	Bar	Gantt	Cube
Candlestick	Open-high-low-close	High-low-close	Pie	Doughnut
Pareto	Pyramid	Radar	Contour/ Spatial	Surface

Color

To change colors on a graph, click the **Color** button on the Graph toolbar. After selecting a color, you can drag and drop that color to any part of the graph.

Grid Lines

To show or hide the vertical or horizontal grid lines on your chart, click the **Vertical** or **Horizontal** button(s).

Legend Box

To display or hide the legend, click the **Legend box** button. You can also move the legend box by dragging and dropping it, and can dock it to any edge of the window.

Data Editor

Click the **Data Editor** button to open a grid of collected values for the graphed checkpoints or transactions in your graph.

Dimensional View

To change the dimensional view of the current chart, click the **3D/2D** button on the graph toolbar.

Rotation

Click the **Rotation** button to access the Properties dialog box where you can change the perspective of your graph. The Rotation button is only available for 3D graphs.

Z-Clustered

Click the **Z-Clustered** button to display multiple checkpoints in separate rows along the Z axis. This button is *only* available for 3D graphs showing more than one checkpoint or transaction.

Tools

Click the **Tools** button to access a menu where you can choose to show/hide the legend box, data editor, graph toolbar, pattern palette, and color palette.

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 and can be sorted and displayed in *QA*Load Analyze in much the same way as *QA*Load 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. Compuware recommends that you view the section "Accessing Test Results" on page 6-1 for general information about how test data is displayed in *QA*Load Analyze before continuing.

Note

ServerVantage resource utilization data is available *only* if you set the ServerVantage integration options on the *QA*Load Conductor's Test Information window before executing a load test. For more information, refer to the *QA*Load Online Help.

Displaying ServerVantage Agent Data

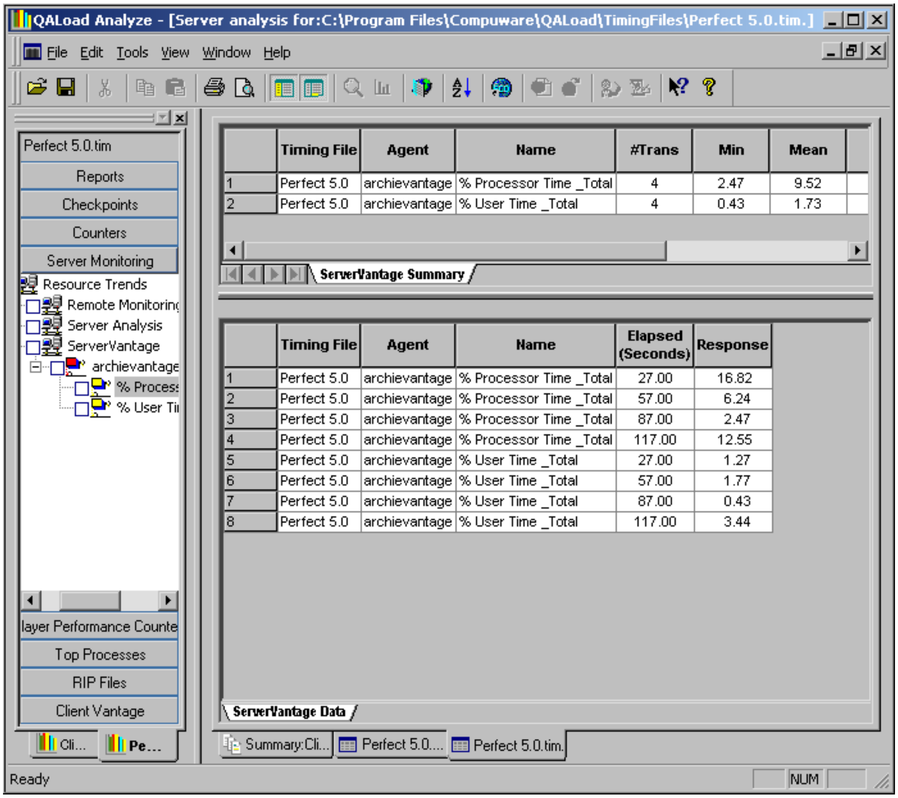
When you open a timing file containing ServerVantage Agent data, *QA*Load Analyze displays test data with *QA*Load timing data two ways:

- 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 (see "Displaying Agent Data Point Details" on page 6-26).
  - Graph Agent data point details (see "Graphing ServerVantage Agent Data" on page 6-27).
- 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.

Displaying Agent Data Point Details

To display details for a specific Agent data point, complete the following steps.

1. In the Server Monitoring group Workspace tree-view, under the *Resource Trends (ServerVantage)* branch, select the Agent data point for which to view details.
2. Click the **View Detail** button or choose **Detail** from the **View** menu to open the ServerVantage detail view in the Data window.



	Timing File	Agent	Name	#Trans	Min	Mean
1	Perfect 5.0	archievantage	% Processor Time _Total	4	2.47	9.52
2	Perfect 5.0	archievantage	% User Time _Total	4	0.43	1.73

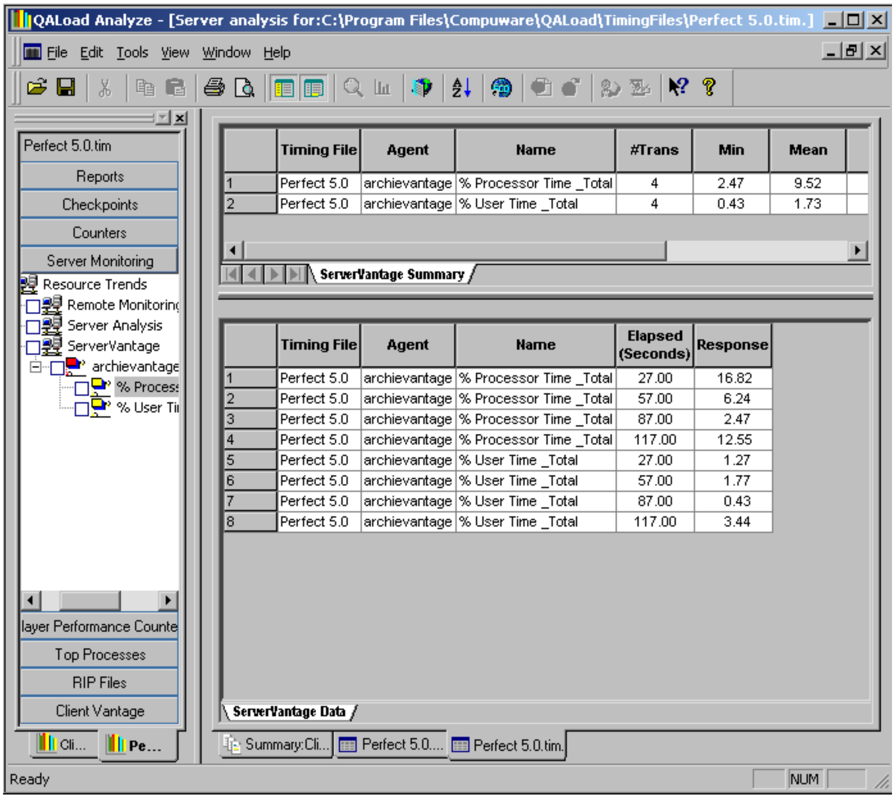
	Timing File	Agent	Name	Elapsed (Seconds)	Response
1	Perfect 5.0	archievantage	% Processor Time _Total	27.00	16.82
2	Perfect 5.0	archievantage	% Processor Time _Total	57.00	6.24
3	Perfect 5.0	archievantage	% Processor Time _Total	87.00	2.47
4	Perfect 5.0	archievantage	% Processor Time _Total	117.00	12.55
5	Perfect 5.0	archievantage	% User Time _Total	27.00	1.27
6	Perfect 5.0	archievantage	% User Time _Total	57.00	1.77
7	Perfect 5.0	archievantage	% User Time _Total	87.00	0.43
8	Perfect 5.0	archievantage	% User Time _Total	117.00	3.44

Graphing ServerVantage Agent Data

Use the following procedure to graph data for one or more ServerVantage Agent data points.

1. In the Workspace tree-view in the Server Monitoring group, under the *Resource Trends (ServerVantage)* branch, select the Agent data point(s) you want to graph.

2. Click the **View Graph** button or choose **Graph** from the View menu. Data for the selected Agent(s) is graphed in a line graph format in the Data window.



	Timing File	Agent	Name	#Trans	Min	Mean
1	Perfect 5.0	archievantage	% Processor Time _Total	4	2.47	9.52
2	Perfect 5.0	archievantage	% User Time _Total	4	0.43	1.73

	Timing File	Agent	Name	Elapsed (Seconds)	Response
1	Perfect 5.0	archievantage	% Processor Time _Total	27.00	16.82
2	Perfect 5.0	archievantage	% Processor Time _Total	57.00	6.24
3	Perfect 5.0	archievantage	% Processor Time _Total	87.00	2.47
4	Perfect 5.0	archievantage	% Processor Time _Total	117.00	12.55
5	Perfect 5.0	archievantage	% User Time _Total	27.00	1.27
6	Perfect 5.0	archievantage	% User Time _Total	57.00	1.77
7	Perfect 5.0	archievantage	% User Time _Total	87.00	0.43
8	Perfect 5.0	archievantage	% User Time _Total	117.00	3.44

3. If necessary, click the **Show/Hide Markers** button on the *QA*Load Analyze toolbar to view data points or use the features available from the graph toolbar to change the way data is displayed in a graph. For information about the features available from the graph toolbar, see "Customizing Graphs" on page 6-23.

Hint

To improve the visibility of any graph, choose **View>Workspace** and/or **View>Workbook** from the *QA*Load Analyze menu to temporarily hide the Workspace and/or timing file tabs. The Data window will expand into the additional space, enlarging the graph.

Where to Go Next

- To convert test data to comma-separated value (CSV) format or HTML for viewing, reporting, or exporting purposes, see "Exporting Test Results to CSV or HTML Formats or RIP files" on page 6-30.
- For information on emailing results, see "Emailing Test Results" on page 6-31.

Viewing Application Expert and QALoad Integrated Reports

*QA*Load integrates with Application Expert version 8.0 and 9.0 to help analyze network performance during a load test. Application Expert is a Windows-based tool that enables users to examine the effects the network will have on the performance of new or modified applications prior to live deployment. Application Expert provides reports that help network managers identify poorly performing applications.

When using the Application Expert integration in a load test, *QA*Load generates a trace file. This file is the capture file created by the Application Vantage Agent.

Note

The trace file extension for Application Expert version 8.0 is .opt; for version 9.0 it is .opx.

The name of the trace file is <Session>_<YYYYMMDD>_<HHMMSS>.opt/.opx where <Session> is the name of the *QA*Load Conductor session used to execute the load test, and <YYYYMMDD> and <HHMMSS> are the date and time the trace file was captured. It is located in the (default) directory **\Program Files\Compuware\QALoad\LogFiles**.

Note

To generate the trace file, Application Vantage Agent must be installed on the same machine as *QA*Load Conductor. The Agent can be installed during the *QA*Load install or installed independently.

At the end of a load test, a high-level static report and various supporting files are automatically generated from the trace file and located in the directory **\Program Files\Compuware\QALoad\LogFiles\<Session>_<YYYYMMDD>_<HHMMSS>**. View the static report at any time in a Web browser such as Microsoft Internet Explorer. It contains the following Vantage views:

- Performance Overview
- Network Utilization and Transit Time
- Node Processing Detail
- Node Sending Detail
- Bounce Diagram
- Error Analysis
- Thread Analysis
- Conversion Map

A description of each view is provided in the <Session>_<YYYYMMDD>_<HHMMSS>.xml file.

Note

To generate the high-level static report, Application Expert or Application Vantage must be installed on the same machine as *QA*Load Conductor.

For additional test analysis, import the trace file into Application Expert or Application Vantage. To use Application Expert or Application Vantage to further analyze the trace, refer to the Application Expert or Application Vantage User's Guides or online help.

Exporting Test Results to CSV or HTML Formats or RIP files

*QA*Load Analyze allows you to convert test data into three convenient formats for viewing or exporting purposes:

- **HTML** — Export the data in a detail view or graph to HTML files that you can conveniently view in a default Web browser or send as attachments in an email message. See "Exporting Test Results to HTML" on page 6-30.
- **Comma-separated value (CSV)** — Export the data in a detail view to comma-separated value (CSV) files that you can import into popular spreadsheet applications. See "Exporting Test Results to CSV" on page 6-31.
- **RIP** — Anytime a user fails during load testing, *QA*Load generates a RIP file containing user errors. If a timing file has RIP file data, you can export a .rip file to the working folder to view within *QA*Load Analyze or *QA*Load Script Development Workbench. See "Exporting RIP Files" on page 6-31.

Exporting Test Results to HTML

The following procedure explains how to export data from detail views or graphs, to HTML.

1. Open a timing file.
2. Generate a detail view or graph.
3. Select the data to export:
    - To export a detail view, click anywhere in the detail view so it is the active view.
    - To export a graph, first be sure that the graph looks the way you want the exported data to look. You may need to maximize *QA*Load Analyze or close the Workspace to make the graph easier to read. Then, click anywhere on the graph to select it.
4. From the **File** menu, choose **Export >Data**. The Save As dialog box appears.
5. Navigate to the location for saving the HTML file to and type a name for the file.
6. Select **Web Page (*.htm, *.html)** as the file type and Click **Save**.

Exporting Test Results to CSV

The following procedure explains how to export the data from the detail view to CSV.

1. Open a timing file.
2. Generate a detail view.
3. Click anywhere in the detail view to select it.
4. From the **File** menu, choose **Export>Data**. The Save As dialog box appears.
5. Navigate to the location for saving the file to and type a name for the file.
6. Select **CSV (comma delimited) (*.csv)** as the file type and Click **Save**.

Exporting RIP Files

The following procedure explains how to export RIP file data to the working folder.

Note

If a timing file does not contain any RIP data, then a RIP Files group will not exist in the Workspace.

1. Open a timing file.
2. In the Workspace, click the RIP Files group.
3. In the tree view, select the appropriate check box for each RIP file to export.
4. Right-click the selected files and choose **Export**. The Browse For Folder dialog box appears.
5. Select the folder to export the RIP file data to. The working folder is the default.
6. Click **OK**. *QA*Load Analyze exports the RIP file to the working folder or the folder designated in step 5.

Emailing Test Results

If you use a Microsoft mail program, send an email message from *QA*Load Analyze containing a timing file or pre-defined reports of test result files as attachments. *QA*Load Analyze will create an email message with the selected files as attachments. The recipient(s) of the message will be able to open the files in a Web browser.

To email pre-defined reports:

1. In the Workspace, select a timing file tab.
2. Select the Reports group.
3. From the **File** menu, choose **Send>Reports**.

*QA*Load Analyze creates an email message with all of the pre-defined reports, .xml, .xsl, and other associated files as attachments. Address the email, add message text, and send the message.

To email a timing file:

With the timing file open, choose **File>Send>Timing File** from the *QA*Load Analyze menu. *QA*Load Analyze will create an email message with a copy of the timing file attached. Address the email, add message text, and send the message.

Index