Performance Testing
IBM MQSeries* Infrastructures

MQTester™ for LoadRunner from CommerceQuest Inc.

www.CommerceQuest.com

Table of Contents

Performance Testing MQ Environments Saves Time & Money

CUSTOMER PROFILE
AT&T Wireless

As a result of growth from mergers and acquisitions, AT&T Wireless needed to provide a large number of new users access to their existing inventory control system. The new load requirement exceeded 10,000 transactions per hour. Their goal was to ensure that the capacity of the back end systems and the MQSeries configuration were optimized.

AT&T Wireless evaluated many testing tools. Since MQSeries was an integral part of the test, AT&T disqualified Rational Software as well as Segue from their tool selection. Only MQTester for LoadRunner filled all the requirements.

CommerceQuest consultants created the proper load testing scripts that allowed AT&T Wireless to run their tests. Multiple tests were run at the Sun Microsystems Systems Center in California to ensure the existing architecture could cope with the extra load.

The test results verified the hardware configuration of the SUN hardware and enabled the fine-tuning of the MQ configuration for maximum performance.

The need for performance testing MQSeries systems and applications is dramatically increasing with the growing demand for IBM MQSeries software, a part of the WebSphere suite of e-business applications. Those who own, or are considering a purchase of any MQ-based product should consider how to stress test the MQ environment.

MQTester from CommerceQuest is the only testing software for IBM MQ products that works with LoadRunner, the industry-leading testing tool from Mercury Interactive Corporation. Easy to install and use, MQTester for LoadRunner is the most reliable solution for predicting system behavior and performance of all MQSeries environments. No complex changes, design requirements or application re-engineering is required.

Scalability, reliability and performance are the most important issues when deploying an enterprise or e-business application. Since the number of users and volume of traffic can change in a matter of seconds, load and system testing are critical steps in maintaining 100 percent uptime.

LoadRunner is test automation software that predicts system behavior and performance by testing an entire enterprise infrastructure including e-business, enterprise resource planning, customer relationship management and custom applications. It does this by emulating thousands of users. Each of these users drives the application with real transactions while LoadRunner measures response times, network delays and both server and application performance.

LoadRunner is the only solution for testing an entire enterprise or e-business architecture, including various types of application logic, security, presentation layer, application layer and network protocols. By implementing MQTester for LoadRunner, you can improve MQSeries application performance and make more accurate predictions of how the system will react to production loads.

Using LoadRunner's Scenario Wizard, you can compose test scenarios to stress your application with hundreds or thousands of desktop, terminal or Web users. Alternatively, you can set up a scenario that shows bandwidth to determine what type of response

time a user will receive over a 56K-modem versus an ISDN, T1 connection or other connection.

CUSTOMER PROFILE
Intelligent Finance

IF.com (Intelligent Finance) used Mercury Interactive's tools extensively during the development phase of its online banking systems. IF.com's back end systems utilize MQSeries to retrieve customer information via messages.

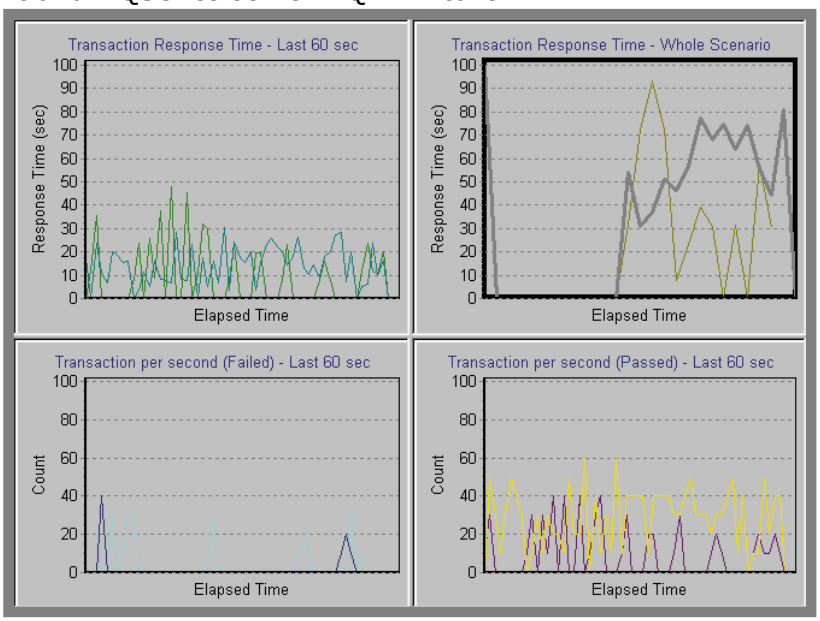
CommerceQuest's MQTester was the pivotal solution chosen to test the MQSeries infrastructure's resilience under the anticipated load generated by the projected number of concurrent requests. The expected peak demand was simulated using MQTester for LoadRunner. The infrastructure's performance was measured to highlight potential problems with unplanned downtime, slow response time, data corruption and the consequential loss of business.

Combined testing of the Web front end and the MQSeries back end enabled IF.com to reschedule its service launch plans to deliver on day one an online bank service capable of exceeding the demands of its customers.

While detecting potential performance problems before deployment is extremely important, just knowing that there is a problem is not enough. LoadRunner's integrated, real-time monitors allow users to isolate performance bottlenecks by splitting the end-to-end transaction response time into separate components of the client, network and server.

For example, the Server Monitor can locate the problems associated with system servers, such as Web servers, application servers or database servers as they become loaded. Similarly, the Network Delay Monitor isolates network performance problems by breaking down the network topology between the client and the server and measuring the network delay between these segments.

At the end of the testing process, LoadRunner creates a series of graphs and reports that summarize test results and presents them in a clear and visual format, as seen below. With MQTester, LoadRunner's Virtual User Generator (VuGen) records MQSeries events, the enhancement of the generated test script, and the play back of the script to emulate conditions of heavy user load. The software records both MQSeries client and MQSeries server MQI API calls.



Before MQTester for LoadRunner

CUSTOMER PROFILE
Amway

In a move to improve their sales, distribution, inventory and warehousing system and plan for growth, Amway decided to move its existing business systems to a new application and hardware platform. Since this change was a bet-your-business move, Amway wanted to be very sure that the new systems and applications could manage the increasing load. In addition, before committing to the purchase of expensive hardware, Amway needed to be very confident that they were ordering the correct model of AS/400 with appropriate capacity.

Since this was a new application, it was impossible to record a test script. Using data message formats, scripts were created manually used to simulate the anticipated system loads. Tests were run at the IBM AS/400 System Center in Rochester, Minnesota on a range of machines of varying capacity.

The information gained from the project provided Amway with the confidence to purchase the AS/400 capable of handling their current requirements as well as providing the hardware capacity that could grow with the business.

Typically, performance testing of MQSeries applications is limited or non-existent. What little performance testing that may be performed is normally conducted only at the single message level. Application developers tend to use single message testing because special test programs require additional development time. Without an actual real-world load on the system, however, single message send-and-receive response time is misleading.

This and other ways of manually testing simple messaging products may have been adequate in the past. However, with the increasing complexity of today's messaging software, manual testing methods are inadequate. Developers are facing short development timelines due to time-to-market pressures, decreasing the time committed to testing. In addition, most MQSeries installations are very large and require sizable teams of programmers, hardware and software designers and test engineers to operate and maintain the software.

Consider some of the limitations of manual testing. Manual testing procedures are difficult, boring and time consuming. Because they are labor intensive, they are visual, user-driven, error prone and expensive. They do not easily support the execution of random test patterns and are inadequate for testing timing errors. They do not easily support the creation of repeatable failure conditions.

Manual testing of messaging applications becomes almost impossible due to the necessity of correlating the load generated with what is happening with the back end application or system. The determination of the type and characteristics of messages passed during the performance test is made even more difficult by attempting to match together data retrieved from different parts of a complex system. Automated testing eliminates these limitations.

With MQTester for LoadRunner

CUSTOMER PROFILE
Southwest Airlines

Southwest Airlines developed a new MQSeries client/server system for passenger check in and seat assignment. The new system replaced an old terminal system with a custom GUI application.

The new application needed to handle an anticipated 7,500 concurrent users. Because the application was mission critical, adequate redundancy and automatic backup were required. Each workstation had a 'twin tailed' configuration with connectivity to both a local and remote server.

Both MQTester Server and MQTester Client were used on this project to test the multiple MQSeries client and server application configurations. Production and back up scenarios were tested.

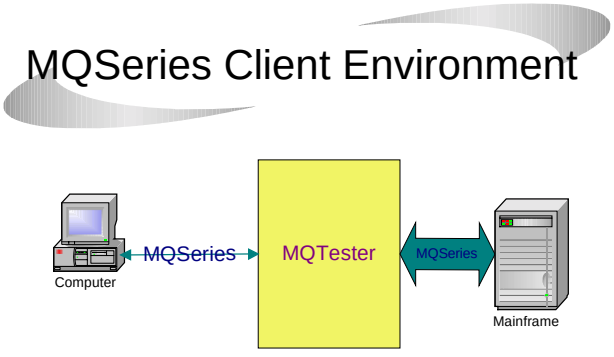
MQTester for LoadRunner was able to test for and reveal a number of design and configuration issues. Southwest was able to quickly make a number of changes and immediately re-test with MQTester. These changes resulted in dramatically improved performance, throughput and reliability. With some final tuning, the application exceeded customer requirements for response time and support of concurrent users.

Southwest Airlines now uses MQTester for LoadRunner as part of the standard quality assurance procedures before any application or configuration changes are made.

MQTester for LoadRunner combines the power of Mercury Interactive's proven automated load testing tool with the WebSphere, MQSeries and MQSI expertise of CommerceQuest software developers and professional services staff. It is the only existing means by which MQSeries systems can be load tested.

MQTester provides both testing and MQSeries professionals the capability to record all of the MQI API calls being issued from the system being tested, including all messages in a business process that a typical user would execute during the workday. By recording the exact MQI calls being issued, the test script recorded by MQTester perfectly simulates the user interaction. Recording the business process allows the performance of a system to be thoroughly evaluated in a real-life scenario.

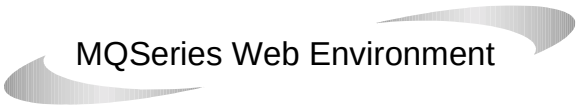
Once a script is recorded, the data being passed in the message can then be parameterized to simulate a normal user interaction with the system. Putting the message data in parameters ensures that the system or application being tested is thoroughly exercised.MQTester fits into the MQSeries environment in multiple locations and the capabilities of MQTester allow for the simulation of multiple environments including client-to-server, server-to-server and Web.

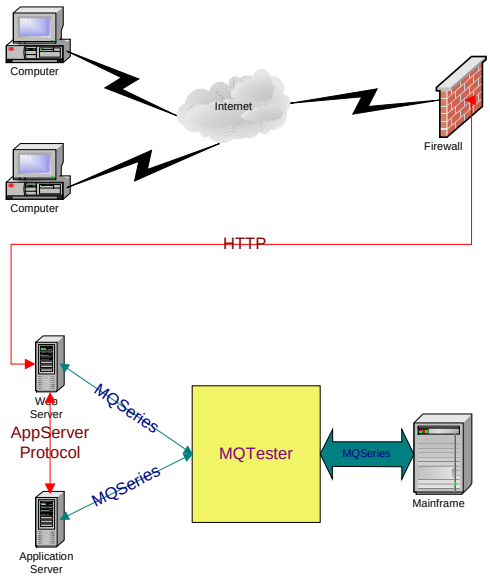


In a client environment, MQTester enables the recording of the MQI API calls being sent and received by the client application. MQTester records all of the MQI calls and builds a script based on the recording.

The contents of all messages sent and received are captured to allow for parameterization of the data.

In a server-to-server environment, MQTester enables you to record the MQI API calls being sent and received by your server application. MQTester records all of the MQI calls and builds a script based on the recording. The contents of all messages sent and received are captured to allow for parameterization of the data. MQTester allows for server configuration and application configuration tuning to optimize their operation.





With MQSeries in a Web environment it is a good idea to separate the testing environments. The standard Web environment requires testing the HTTP protocol going to and from the Web server to client browsers. When testing the Web environment using Web virtual users, it is sometimes difficult to determine where the performance problems are located. By using MQTester along with Web virtual users, it

becomes easier to determine which part of an application is causing performance problems.

How it Works

MQTester for LoadRunner Features

Version 1.0

Record MQSeries Client MQI API calls (Windows NT)

Record MQSeries Server MQI API calls (Windows NT)

Replay MQSeries Server MQI API calls (Windows NT)

Put variables in the contents of the outgoing messages (this allows the replaying script to act more like a real user)

Multiple MQTester Virtual Users (Vusers) executing on a Vuser host machine

Testing of any infrastructure supporting MQSeries

Version 2.0

Replay MQSeries Client MQI API calls (Windows NT, Sun Solaris, HPUX, AIX)

Record Visual Basic-based MQSeries application (Client and Server MQI API)

Threaded Vuser replay

Formatted data file based on known message structures

Testing Process Initiation

1. Invoke LoadRunner's VuGen to record the basic script. MQTester writes the MQ calls to the following files:
   - *Log file* generates a test script
   - *Data file* saves the actual data exchanged between the application and the queue manager
2. Enhance the test script as needed:
   - Define parameters and correlate statements as needed
   - Configure the runtime settings
3. Replay the script to test the system.

Record Component

1. The record component records the MQSeries calls made to the MQSeries queue manager. The recorder logs the output in two files:
   - The MQ calls are output to the **MQ_Recording.log** file
   - The actual data exchanged between the application and the queue manager is output to a **data.mq** file in ASCII format
2. During recording, MQTester intercepts the MQI API function call and records all of the parameters passed to it. The message data is written to the data.mq file. The MQI function name and all parameters are written to the MQ_Recording.log file. The recorder maps the MQSeries function calls to those of the replay component and logs the corresponding replay component functions.

Script Generation Component

The script generation component reads the MQSeries function calls that were recorded in the MQ_Recording.log file and then generates the files that form the test script.

Replay Component

The replay component implements the functions written by the script generator. Those functions include common utility functions for buffer and variable manipulation plus MQSeries-specific functions.

Harness the Full Power of MQSeries with Enterprise Testing

CommerceQuest's three-way partnership with IBM and Mercury Interactive has produced the world's first MQSeries application performance testing solution.

IBM and CommerceQuest work in close partnership to provide essential business solutions based on message-oriented middleware. IBM MQSeries, the most widely used message-queuing software on the market, enables users to exchange information between applications across more than 35 different platforms, from mainframes to PCs. This award-winning software enables business integration throughout the enterprise, allowing companies to maximize e-business opportunities by leveraging existing resources to improve speed-to-market and anticipate IT changes as their business changes.

CommerceQuest is repeatedly honored as an award-winning IBM business partner. Its awards include IBM Solution Excellence for e-Commerce, MQSeries Partner of the Year, IBM Software Investment Partnership Top Contributor, and IBM Training Provider Award of Excellence. CommerceQuest's software has achieved IBM's *Cluster Proven* certification.

CommerceQuest offers extensive MQSeries, MQSI and WebSphere professional services as well as a software development practice specializing in solving the problems of enterprise and business-to-business integration. CommerceQuest integration solutions including data movement and application integration software, component-based development frameworks, MQTester, and the MQSeries Client for 4690 dramatically accelerate the return on investment IBM customers receive from their WebSphere and MQ investments.

Mercury Interactive is the leading provider of Web performance management solutions that help e-businesses ensure a positive user experience. As a participant in Mercury Interactive's Open Test Architecture™ program, CommerceQuest developed MQTester and with it extended LoadRunner's support to include MQSeries

environments. Mercury Interactive solutions turn application performance, scalability and user experience into competitive advantage. The company's performance management products and hosted services are open and integrated to test and monitor business-critical applications. CommerceQuest's Mercury-certified product specialists and instructors assist enterprises with all phases of the test cycle from the planning of automated functional and performance testing to hands-on application testing.

Summary

MQTester for LoadRunner allows you to harness the leading testing technology of Mercury Interactive for your MQSeries applications. With MQTester, all of LoadRunner's scripting, playback, monitoring, reporting and analysis capabilities are extended to the MQSeries infrastructure for extensive testing and prototyping to be performed by testing professionals and MQSeries professionals alike. With MQSeries growing importance in the enterprise and e-business world, testing MQSeries infrastructures is as important as testing the applications that use it.

About CommerceQuest

Companies like Anheuser Busch, American Express, Ericsson and EDS trust CommerceQuest, an award-winning integration solutions provider and IBM Business Partner, to execute their business interoperability strategies. CommerceQuest's software and professional services drive dramatic improvements in business flexibility, effectiveness and efficiency for companies around the world. With CommerceQuest, businesses bring their e-commerce projects on line quickly, minimizing risk, cost and time, while maximizing business value. From legacy applications to the Web, CommerceQuest software and services make complex business integration work. Any data, any platform, any network — every time.

A member of Internet Capital Group's (Nasdaq:ICGE) collaborative network of Partner Companies, CommerceQuest was founded in 1992 and is privately held and headquartered in Tampa, Florida. The company's integration software and services are available throughout the United States, Pacific Rim and Europe. For more information please visit www.CommerceQuest.com.

** Indicates trademark or registered trademark of International Business Machines Corporation. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.*