



High-ROI MQ Testing using Mercury LoadRunner & the Mentora MQ Accelerator™



IBM WebSphere MQ

- **Organizations use IBM WebSphere MQ**
 - As a backbone for integrating loosely-coupled systems using publish-and-subscribe messaging distribution
 - To connect applications implemented in web services and 80+ supported platforms
 - To enable transaction management and secure file transfers
- **MQ is revenue-critical for transaction-dependent industries**
 - Financial services: Electronic transfers, ATM transactions
 - Transportation: Reservations systems
 - Manufacturing: Supply chain orders and fulfillments
 - Healthcare: Claims submission and payment
- **Once implemented, ensuring performance is critical**

But how do you test? IBM provides no standard performance testing tools!

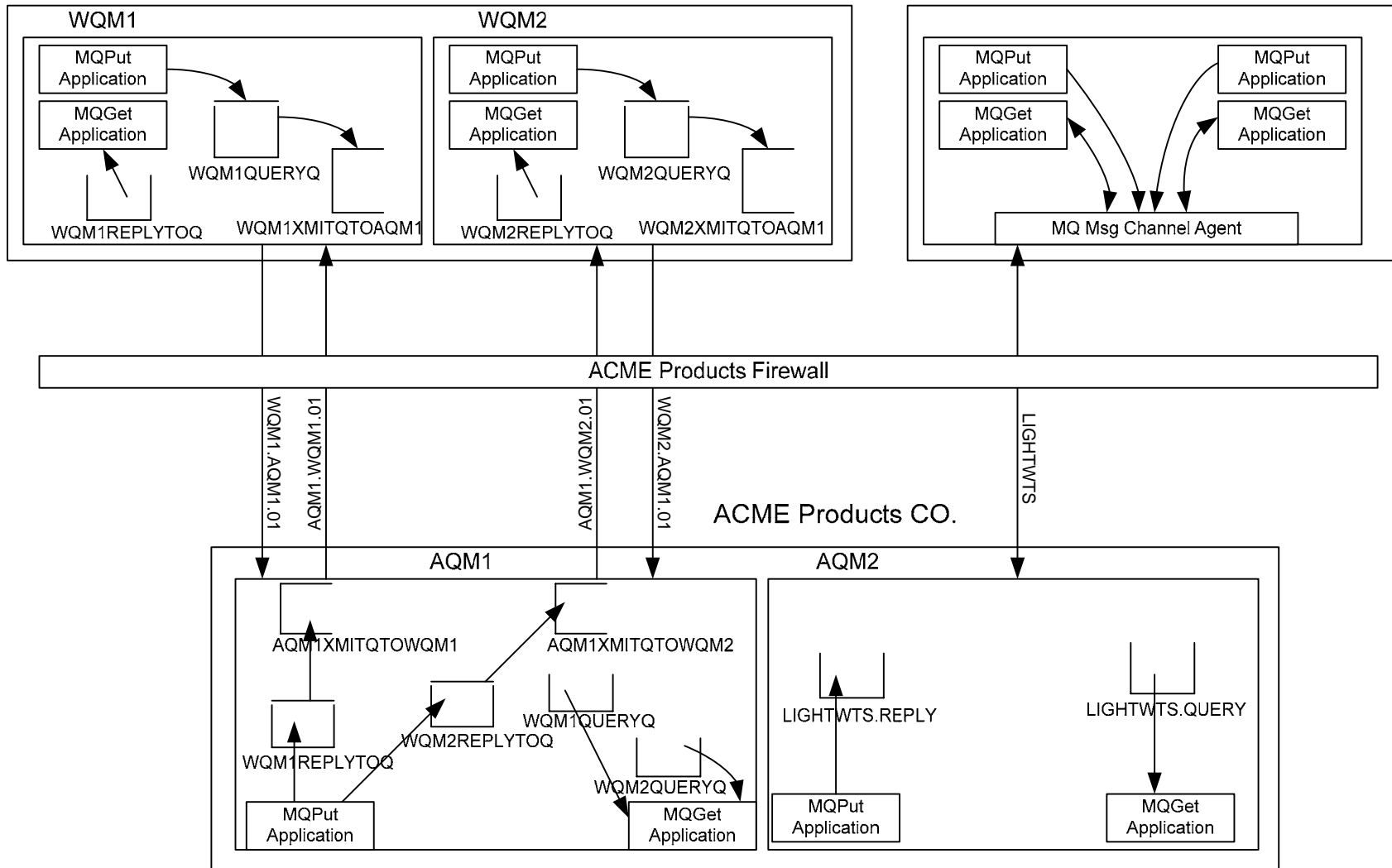
MQ Performance Testing Needs

- 1. Confirm that the system works**
 - Assure messaging transactions are delivered where they are expected
 - Confirm exception processing works as designed
- 2. Assess performance metrics for capacity planning**
 - Verify load limits of hardware and software for capacity planning
 - Validate optimum queue and channel parameters for high performance
- 3. Perform MQ application load testing**
 - Assure applications meet high-availability standards under stress
 - Identify potential bottlenecks before they occur in production
- 4. Ensure quality of business transaction processing**
 - Measure transaction delivery time against 'SLA' response times
 - Report 'end-user' anomalies

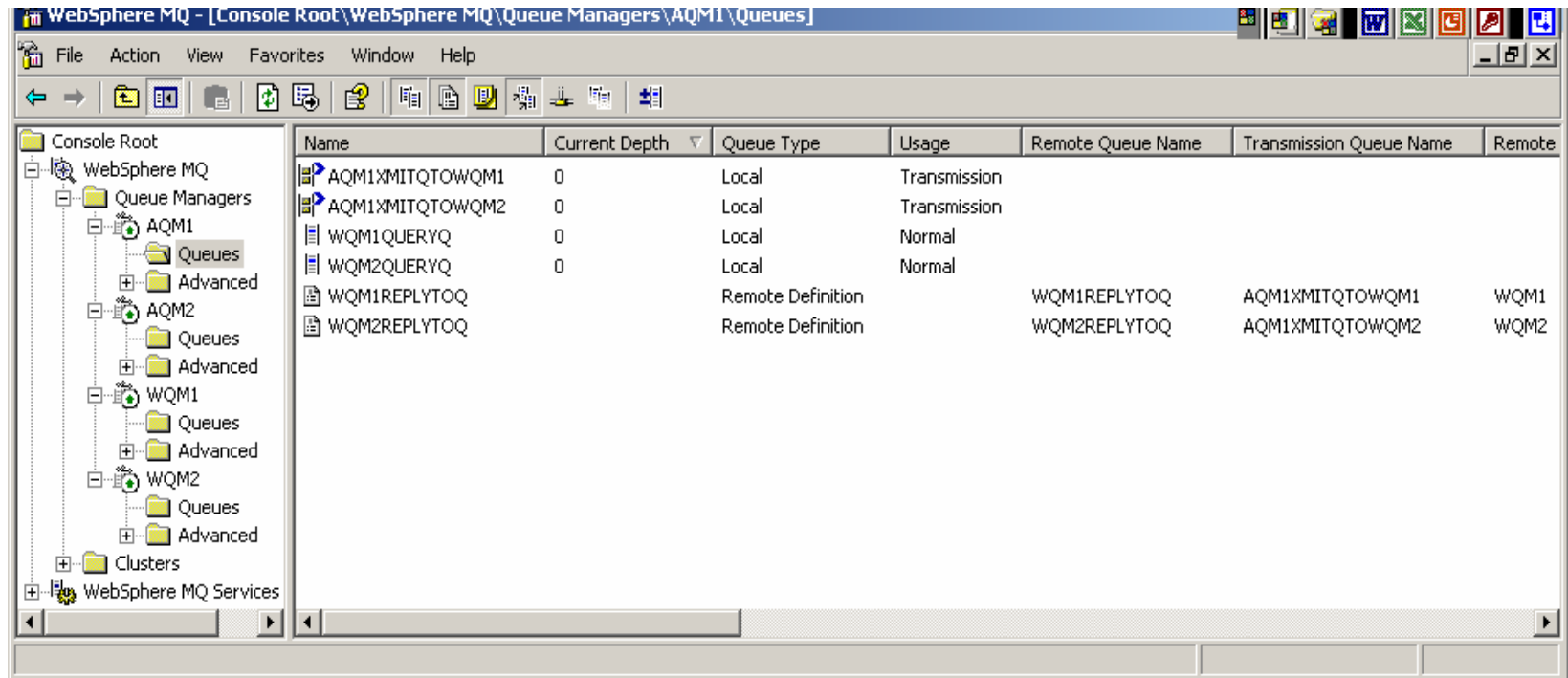
Context: Typical WebSphere MQ Architecture

Widgits Inc.

Lightweights.com



Context: Typical MQ Components



The screenshot shows the WebSphere MQ console interface. On the left is a tree view of the console structure, including Console Root, WebSphere MQ, Queue Managers, and various Queue Managers (AQM1, AQM2, WQM1, WQM2) with their respective Queues and Advanced settings. On the right is a table listing the queues.

Name	Current Depth	Queue Type	Usage	Remote Queue Name	Transmission Queue Name	Remote
AQM1XMITQTOWQM1	0	Local	Transmission			
AQM1XMITQTOWQM2	0	Local	Transmission			
WQM1QUERYQ	0	Local	Normal			
WQM2QUERYQ	0	Local	Normal			
WQM1REPLYTOQ		Remote Definition		WQM1REPLYTOQ	AQM1XMITQTOWQM1	WQM1
WQM2REPLYTOQ		Remote Definition		WQM2REPLYTOQ	AQM1XMITQTOWQM2	WQM2

Management Console for configuring and administering your MQ environment

Test Requirements

- **Queue Manager to Queue Manager Performance**
 - Message queue depths
 - Channel capacities by message type/size
 - MQ Application capacity limitations
 - Server resource utilization
- **Messaging clients**
 - Message sends (GETs)
 - Message responses (PUTs)
 - Client channel connections
 - Performance of MQ client applications

Test Requirements - 2

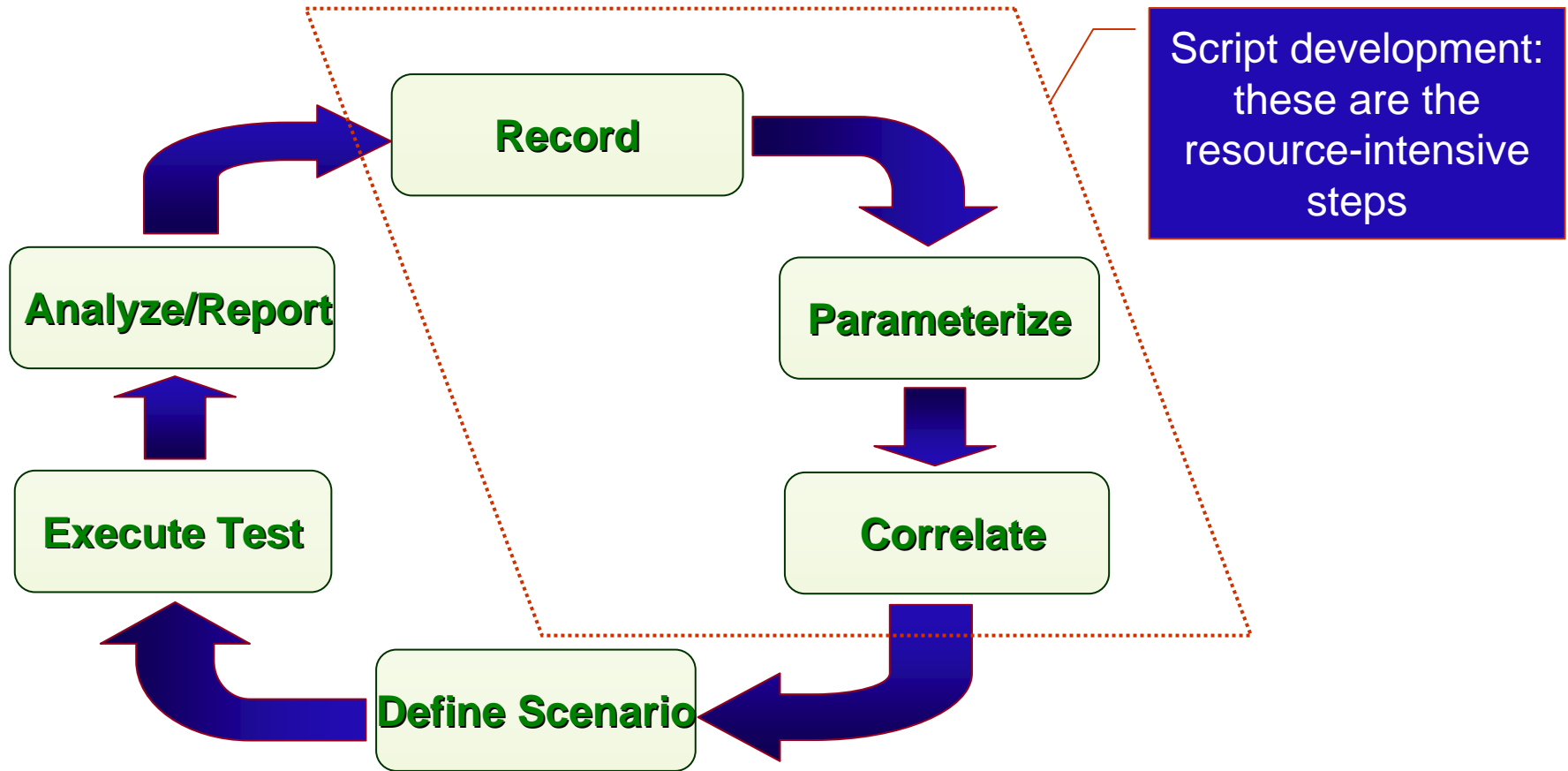
- **Cluster Queue Managers and/or Queue Sharing Groups**
 - Workload balancing
 - Failover processing
 - High availability environment
- **MQ Infrastructure bottlenecks**
 - Message delivery delay points
 - Multi-threaded applications
 - Queue Manager and/or System failure impacts

Testing Solutions

- **“Roll your own”**
 - Some MQ architects* say they’ve developed their own test harnesses for testing MQ – with varying effectiveness
- **Mercury LoadRunner**
 - A few MQ architects* say they use Mercury LoadRunner, coupled with their specialized MQ vuser protocol
 - They develop new scripts for each new component / configuration

* From survey at MQ BOF at the June’06 IBM Transaction & Messaging conference

MQ Testing Process with LoadRunner



MQ Testing Challenges with LoadRunner

- **Queue Manager and Client configurations are unique***
 - Name, port, message ID, persistence, expiry, message size & type, etc.
- **LoadRunner MQ vuser script recording is monolithic**
 - Each Queue manager combination requires its own script with specific configurations
 - Results in many scripts to develop, test, debug, manage
- **High overhead by “standard” MQ VUser**
 - Message files typically 2K – 200K bytes, resulting in high VUser memory footprint; thus few VUsers per megabyte of memory on load generator

* And there are a multiplicity of combinations in the typical MQ environment.

Vanilla LoadRunner Testing: Low ROI

Activity	Duration* (hours)
Typical script development time**	8
Typical script maintenance time per test iteration	2
Typical number of scripts per Queue Manager	10
Total initial development hours	80
Total maintenance hours per test iteration	20

* Data from an MQ Solutions expert at a worldwide travel distribution company

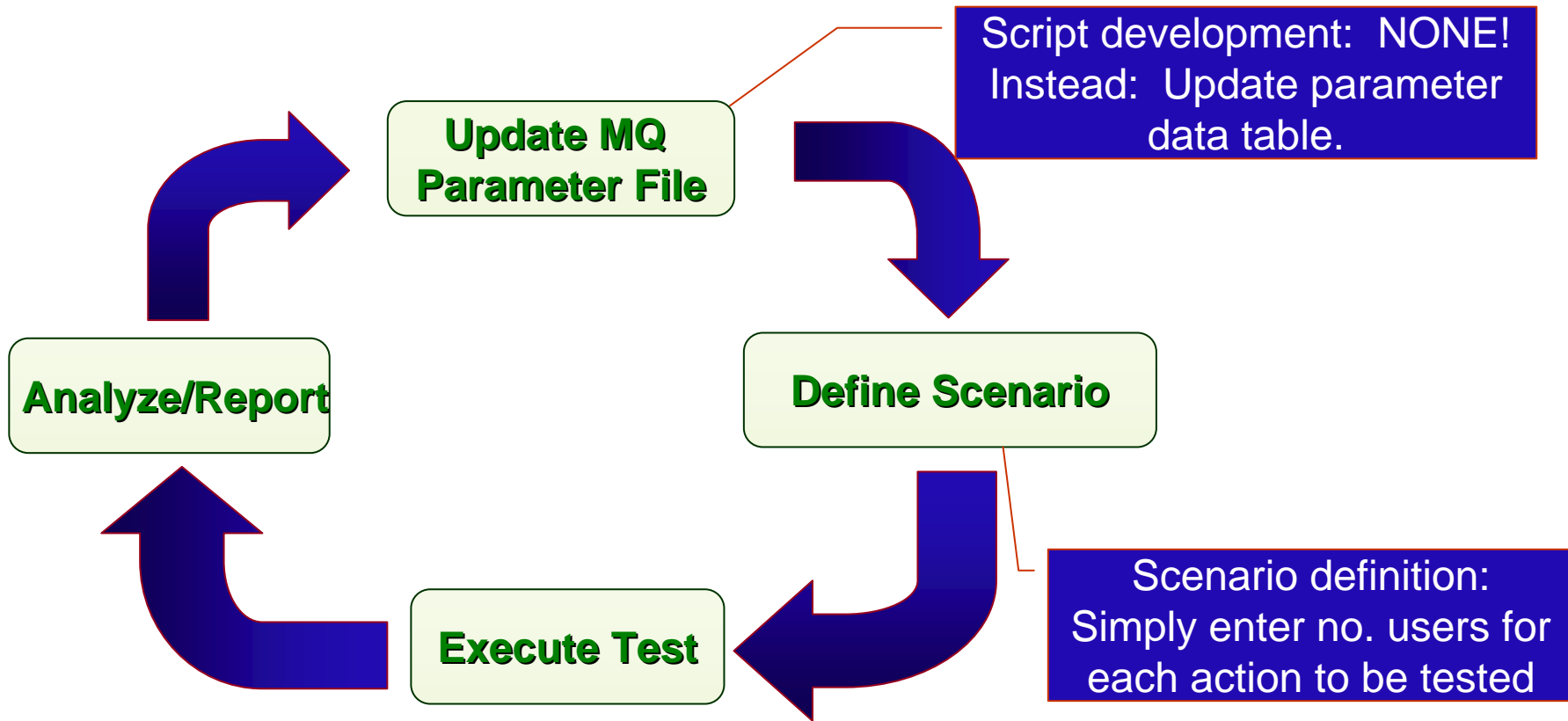
** Assumes expertise with both LoadRunner and MQ

Low reusability, Low ROI

Solution: Mentora MQ Accelerator

- **What it is**
 - A LoadRunner add-on
 - A single, configurable LoadRunner MQ template script
 - Utilities for installation, configuration and execution
- **Features**
 - Parameters for all key values
 - Embedded transaction markers to capture key performance metrics
 - High degree of granularity using unique dynamic transaction names
 - In-memory shared message file to reduce VUser memory footprint
 - Self-regulating load during test execution based on queue depths
 - Queue cleanup capability following a test execution
 - Installer adds all needed options to LR config files

MQ Testing Process with MQ Accelerator



**Spend your time on the high value activities:
results analysis and bottleneck resolution**

MQ Accelerator: Benefits

- **Reduced time-to-first-test**
 - ZERO scripts to develop
 - ONE script, configurable for each MQ application to be tested
 - ***1 hour to first test!***
- **Low/No script maintenance cost**
 - Once defined, no maintenance required: upon MQ application or object changes, simply modify parameters and re-test
 - Predefined LR Controller Scenario file
- **High ROI**
 - Low-cost initial investment
 - Cost recouped in first project

MQ Accelerator Script

The screenshot displays the Mercury Virtual User Generator interface. The main window shows the script content for 'ClientPutMQTesterFrame1 - MQSeries Client'. The script includes initialization functions and a list of include files. A blue callout box highlights the script's origin. Below the script, the execution log shows the start of the virtual user script and various configuration parameters.

MQ Accelerator script viewed in LoadRunner VuGen 8.1

```
ClientPutMQTesterFrame1 - MQSeries Client
vuser_init
Action
vuser_end
data.mq

// User Initialization Function
//
// Generated by MQTester for LoadRunner
// MQTester for LoadRunner developed by CommerceQuest, Inc.
// Copyright (c) 2000 - 2003 CommerceQuest, Incorporated.
// All rights reserved.
//
// LoadRunner MQ Script
//
// Generated by Mentora MQ Accelerator
// Copyright (c) 2006 EfficientLK, LLC.
//
// Script Initialization section
//
#include "lrmq.h"
#include "vuser_init_declarations.h"
#include "Externals.h"
#include "GenericFunctions.h"
#include "MQLRServiceFunctions.h"
#include "OpenQueues.h"
#include "ScriptTypeValidation.h"

Execution Log
Recording Log
Virtual User Script started
Starting action vuser_init.
GenericFunctions.h(11): ScriptPath: C:\MQExelerator\Scripts\ClientPutMQTesterFrame1\ClientPutMQTesterFrame1.usr
ScriptName: ClientPutMQTesterFrame1
USRfileNameLenght is: 821
[General]
Type=mqseries-client
DefaultCfg=default.cfg
AppName=
BuildTarget=
```

MQ Accelerator Script Parameters

MQ Accelerator script parameters enable full configurability to match your MQ environment

Parameter List

Parameter type: File

File path: MQAcceleratorMainConfig.txt

Add Column... Add Row... Delete Column... Delete Row...

	ScriptName	roundtrip	Qmgr	rQmgr	LastItem
1	ClientPutMQTest	N	AQM1	WQM1	end

Edit with Notepad... Data Wizard...

Select column:

By number: 1

By name: ScriptName

File format:

Column delimiter: Comma

First data line: 1

Select next row: Sequential

Update value on: Each iteration

When out of values: Continue with last value

Allocate Vuser values in the Controller:

Automatically allocate block size

Allocate [] values for each Vuser

New Delete

Close

MQ Accelerator Load Scenario

The screenshot displays the Mercury LoadRunner Controller interface for a scenario named "MQAcceleratorDemo". The "Scenario Schedule" section shows the following configuration:

- Schedule Name: Default Schedule
- Mode: Scenario Scheduling
- Scenario Duration: Until Completion
- Load Behavior: Load all Vusers simultaneously

A "Load Preview" graph shows a constant load of 30 users over time. A blue callout box states: "Pre-configured MQ Accelerator Controller scenario configurable for your target load and test environment".

The "Scenario Groups" table lists the following components:

Group Name	Script Path	Quantity	Load Generators
clientget	I:\Lenny\ClientGet	2	db4
clientgetr.1	I:\Lenny\ClientGetRT	2	triangle
clientgetcleanup.1	I:\Lenny\ClientGetCleanup	1	localhost
clientput.1	I:\Lenny\ClientPut	5	db4
clientputr.1	I:\Lenny\ClientPutRT	5	localhost
serverget.1	I:\Lenny\ServerGet	2	triangle
servergetr.1	I:\Lenny\ServerGetRT	2	localhost
serverput.1	I:\Lenny\ServerPut	5	db4
serverputr.1	I:\Lenny\ServerPutRT	5	triangle

The interface includes a "Start Scenario" button and other controls on the right side. The bottom status bar shows "Auto Load Analysis" and "Auto Collate Results" options.

MQ Accelerator Pricing

- **One-time, low license fee includes**
 - MQ Accelerator product*
 - A 1-hour on-line tutorial to teach you how to use it
- **Annual maintenance includes**
 - Maintenance release updated, tested and certified on new LoadRunner and MQ maintenance release
- **Optional consulting**
 - Remotely, at a low hourly rate
 - On-site, at a reasonable daily rate (plus expenses)
 - Fully-outsourced implementation project to test your environment can be quoted to meet your needs and timeframes

Contact us at 866-MENTORA for pricing!

* LoadRunner & SOA/Messaging vuser protocol bundle required and licensed separately