Welcome to QEngine

About This Guide

This evaluation guide concentrates only on the Web Performance test tool of QEngine. This will help you to get started and have a smooth experience of using QEngine during your evaluation process. Most of the general queries that arise during the evaluation phase are given in the How Tos section.
If you have any queries/feedbacks regarding QEngine please send us your queries to qengine-support@manageengine.com or post your queries in the forums.

About QEngine

QEngine is a powerful tool for automated functional and performance testing of your web applications and web services. QEngine Web Performance test tool offers the following features:

- HTTP/HTTPS Protocol Support.
- Provision to record and play AJAX requests.
- Modem Simulation to simulate users with different bandwidth.
- Server Monitoring to monitor CPU usage and memory usage for web servers. (in Windows/ Linux).
- Database Monitoring feature to monitor the database parameters for databases such as MySQL, Oracle, etc. Monitors thread details, connection details, query statistics etc.
- Session Tracking (Cookies / URL Rewriting).
- Parameterize any parameters using Response of previous URL / Cookies / Java Script / Dataset (CSV , Database, Excel)
- Support File Uploading in load testing.
- Option to download images / css / javascript by parsing the response of URL.
- Centralized coordination and reporting of distributed load test results.
- Dynamic graphs generated during load test execution.
- Realistic Load Scenarios (Normal/Ramp-up/Ramp-down/Burn-in).
- Correlation (substituting values in dynamic data while playback).

- Basic/Digest/NTLM Authentication Support.
- Summary and Detailed Reports/Graphs.
- Virtual User Simulation with a provision to assign unique IP to each user.
- Schedule regression testing to run the test in unattended mode.

Online Resources

QEngine includes the following online resources:

- **What's New**: Provides the latest features and enhancements in the current version of the product.
- **Product Documentation**: Provides the complete user guide for all the test tools in QEngine in HTML, ZIP and PDF format.
- **Features Document**: Provides the complete list of features with screenshots for all the test tools in QEngine.
- **Datasheet**: Provides the features with screenshots in PDF format for all the test tools in QEngine.
- **Comparison Documents**: Provides the comparison documents for Web Functional, Web Performance and Web Services test tools of QEngine.

Installing/Starting QEngine

Installing QEngine

- **Installation in Windows**

- To install QEngine in Windows, download the EXE file **ManageEngineQEngine.exe** from the site http://www.manageengine.com/products/qengine/index.html and install it by executing the EXE file.

- **Installation in Linux**

- To install QEngine in Linux, download the BIN file **ManageEngineQEngine.bin** from the site http://www.manageengine.com/products/qengine/index.html and execute it.

Uninstalling QEngine

- **In Windows**

    - QEngine can be uninstalled through the OS Control Panel Add/ Remove Programs options.

- **In Linux**

    - QEngine can be uninstalled by removing the directory containing the files and directories extracted during installation. At the command prompt, type
        - **rm -rf ManageEngine/QEngineWebTest** to successfully uninstall QEngine.

Starting/Shutting down QEngine

- **Starting QEngine Server**

- After installing QEngine, execute the **StartWebTestServer.bat/.sh** file in <QEngine_Home>/bin directory to start the QEngine server. The server will be started in the default server port 7001.

- **Shutting down QEngine Server**

- To shutdown QEngine server, execute the **ShutDownWebTestServer.bat/.sh** file in<QEngine_Home>/bin directory.

Installing QEngine Toolbar

The QEngine Toolbar will be installed by default while installing QEngine. While connecting the QEngine from different machine then QEngine toolbar needs to be installed. When there is no toolbar installed in the browser with which you are accesing QEngine, it will prompt you to install the QEngine toolbar. Just follow the button link to install the toolbar for your browser.

Connecting to QEngine

After starting the QEngine, the QEngine icon will be shown in the System tray in orange and red color. Right click over the icon and select the "**Start Web Client**" option to lauch the browser and connect to QEngine. This will launch the browser with QEngine page loaded. Thus you can start creating the automation script for your application.

If you are connecting to QEngine from different machine, open the browser and browse to **http://<QEngine_installed_Machine_Name>:7001**. This will take you to QEngine page.

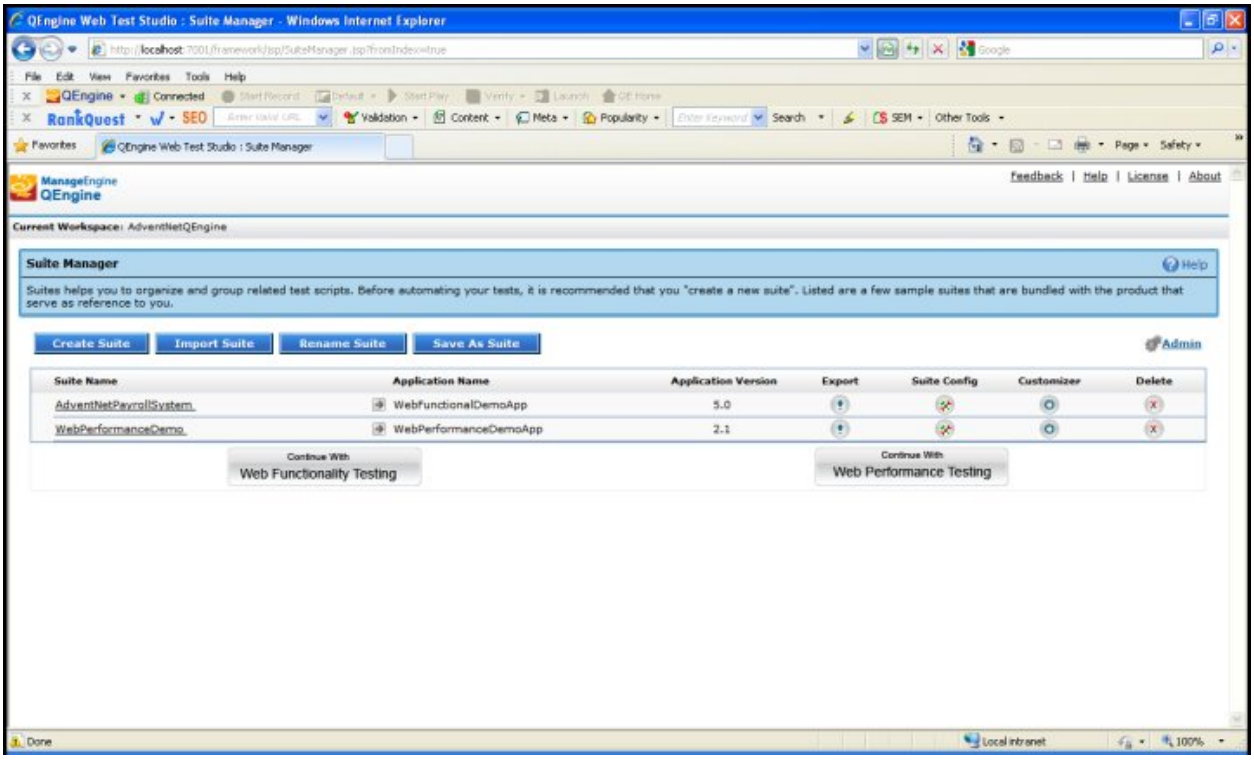
Exploring QEngine Web Performance UI

Organizing Test Scripts using Suite Manager UI

The Suite Manager is used to organize the suites created with QEngine and provides options to choose different test types to create test cases.

Creating Test Suite

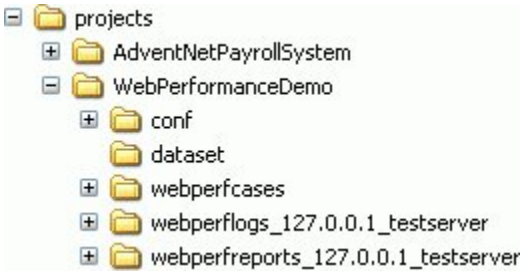
In the SuiteManager page as shown below, select the required suite from the **Available Suites** table or create a new suite by clicking the **Create Suite** button. Suite Manager also provides functionality such as **Import Suite, Rename Suite, Save As Suite**.



For each suite in the **Available Suites** table, you can use the export option to export a suite as a .qed file, configure the suite details such as, proxy details, severity, bug tracker, mail server details and reports using the Suite Configuration UI, set filters to execute specific scripts and delete a suite.

Understanding QEngine Web Performance Test Suite Structure

The recorded load test scripts are created under a defined test suite. QEngine Web Performance Test Suite has the following structure:



WebPerformanceDemo	The Suite name
conf	Contains configuration files required to execute the Suite.
dataset	Contains the CSV files and datasource files to include data-driven load test scripts wherein the values are dynamically fetched from a CSV file or database.
webperfreports	Contains the reports generated during load test execution to analyze the performance bottlenecks.
webperflogs	Contains the logs created while executing the web performance test scripts
webperfcases	Contains the scripts recorded for web performance testing

Selecting Test Type and Creating Load Test Scripts

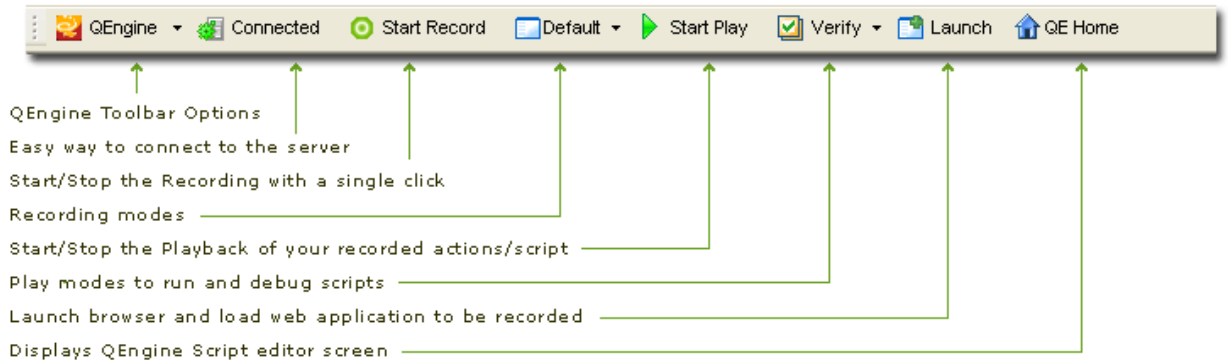
Click on the **Continue with Web Performance Testing** button to go to the web performance test creation screen.

This will bring up the Web Performance Test creation Screen to create / configure / execute performance test for your Web applications. In the test creation screen that gets displayed, click **New Load Test** from the Quick Links placed below the tabs to create a new load test.

Exploring the Main Web Performance UI Options

Exploring QEngine Toolbar Options

The function of each toolbar option available in QEngine ToolBar is explained in the table below :



Exploring Quick Links Options

The function of each quick link option is explained in the table below:

Quick Link Options	Description
New Suite	Invokes new suite creation screen. Suites can be used to organize the load test created
Open Suite	Opens the selected suite.
New Load Test	Show a div to get the load test name and create new load test. You can add transactions to it and configure the users for the load testing.
Instant Load Test	Show a div to get the URL which should be load tested and number of virtual user to be simulated. It will create a load test instantly and execute the load testing.

Exploring Load Test Screen

The functions of each load test options are discussed below:

Load Test Options	Description
New Transaction	Transaction contain sequence of steps that needs to be performed by the virtual users during load testing. New Transaction link is used to create new user scenario in the application.
Import Transaction	Import the already created transaction to perform the new load test with varying parameters / virtual users.
Record More	Record more scenario to the already existing transactions.
Parameterize	By default QEngine take the value for parameters from the response during load test playback. To simulate unique value for each virtual user parameterize it and fetch data from data source such as CSV/DB/XLS.
Rename Transaction	Rename the already existing transaction from the load test.
Delete Transaction	Delete the already existing transaction from the load test.
Load Details	Configure the number of virtual users to be simulated and the type of load to be applied during load testing.Also configure Exit Criteria settings to exit test during some unexpected scenario in the application.
Schedule for Regression	Scheule the current load test to be executed in regression at the specified time.
Modem Simulation	Provide the reports for the load testing based on the configured modem speed

Server Monitoring	Monitor the server for the CPU and Memory usage during load testing
DB Monitoring	Monitor the DB Metrics during load testing
Start Play	Start the load testing for the selected load test.
Settings	Configure the load test settings for the load testing playback.

Understanding Load Testing Process

Objective:

- Planning a Load Test:
    - Identify System Configurations
    - Identify Key User Scenarios
    - Identify Workload
    - Identify Metrics
    - Identify Test Cases
- QEngine Load Testing Process:
    - Create Load test cases
    - Record User Scenarios as Transaction
    - Parameterize the Transaction
    - Assign % of Workload for the users
    - Configure Load details
    - Run Load Test Cases
    - Analyze Test Results
- Following are the steps that will guide to plan your load test and know the details of how to proceed with the load testing process to test your web application.

Planning a Load Test:

Step 1: Identify System Configurations

Identify and setup the system configurations in which your web application should run to measure the performance statistics. For example, your web application running in two machines PIII 1 GHz 512 MB RAM and PIV 2 GHz 2 GB RAM. This will help you to compare two system configurations and identify which system configuration performs better and meets the expected performance criteria in which you can host your web application.

Step 2: Identify Key User Scenarios

Identify the key user scenarios or user transactions that a real-world user would perform and that are critical for your web application from a performance perspective. User scenarios include multiple application activities. Key user scenarios are identified based on specific performance goals such as transactions that consumes time or those that have a significant impact on your application's performance. For example, determine the approximate time required to process a user transaction and return the results .for. e.g., processing the credit card details and returning the status takes approximately 5 seconds, etc.

Some of the key user scenarios are as follows:

- Log into the application, browse the shopping cart details and log out of the application.
- Log into the application, purchase items from the shopping cart and log out of the application.
- Log into the application, search for required items and log out of the application.

Step 3: Identify Workload

Identifying the workload determines the amount of load to be placed on your web application and how it is going to be placed over the test duration time. Decide on the type of load to be placed:

- Whether you need to place a constant number of virtual users for the duration of the test (load testing) to determine the minimum configuration under which the web application can perform satisfactorily. or
- Follow an approach to ramp up the test and then ramp down (peak testing) - say you ramp upto 100 users and remain in the peak state for 30 minutes and then drop the number of users to ramp down at every 60 seconds. This will help you to find the maximum performance limit of your web application. or
- Choose the load type where you run the load test over long periods of time with normal user loads and exit the test when the specified criteria is met. This will help you to assess the web application's stability over long periods of time.

For each user scenario, identify the following:

1. **Number of users** - The total number of users accessing your web application in a given time frame.

2. **Percentage of users** - Identify the percentage of users accessing parts of your web application or the users to be associated with each user scenario / transactions such as 60% of users browsing the home page, 20% of users purchasing items from the shopping cart and the other 20% of users searching for required items, etc.

Step 4: Identify Metrics

Identify the relevant metrics that you need to collect in your load test. This will help you to easily analyze the potential bottlenecks in your web application in relation to your performance objectives. Metrics collected could be such as, CPU usage, memory usage, hits per second, server response time, page download time, error percentage, etc.

Following are the performance metrics that can be monitored:

- **System-specific metrics**: This set of metrics helps you to measure the resource utitlization in your web server. The system resources, such as total % of CPU usage and memory usage of the configured system can be measured.
- **Web application-specific metrics**: This set of metrics helps you to measure the server response time (time between Request sent to time of receiving First Byte), page download time, errors in web pages, throughput (which measures the data transferred relative to some unit of time), etc. You can also measure the overall application throughput, hits per second, response time, page download time, etc.
- **Database-specific metrics**: This set of metrics helps you to measure the database issues. Metrics collected for MySQL include: Thread details, Connection details, query details, table related statistics, etc.

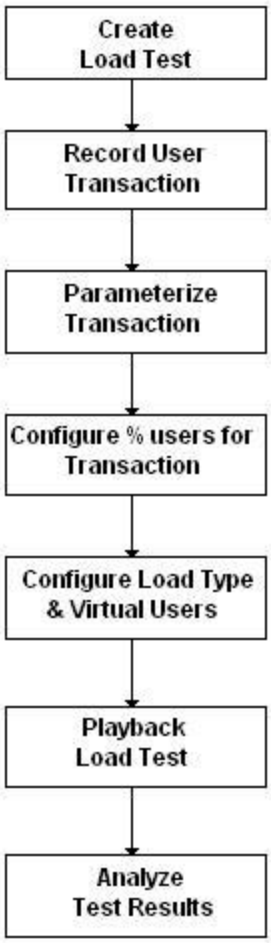
Step 5: Identify Test Cases

Identify and draft the test cases to be load tested. Test cases are identified by deciding which user scenarios identified in Step 2 should be associated with which workload identified in Step 3.
Once you identify the above, create the load test case and run the test for multiple iterations, analyze the results, use the results to improve both the subsequent testing and/or tune the system resources or your web application being tested, and run the load test again.

QEngine Load Testing Process

Once you plan your load test, the next step is to perform the load testing process. Load testing is the process of identifying performance bottlenecks in your web application under normal and peak loads. This will help you to tune the resources (both the hardware and the software) of your web application and optimize the user experience for maximum performance.

Following are the steps involved in QEngine Load Testing process:



Step 1: Create Load Test

Create a load test which will contain the user transaction and load configuration for the load test. In addition, you can configure **Server Monitoring / DB Monitoring** in the load test to monitor and report the respective server CPU and Memory Usage during the load testing. This will be useful to identify the server capacity required.

Step 2: Record User Transaction

Create a **New Transaction** and record the user scenarios in your web application. Here, the user scenarios are the one identified in your planning phase in **Step 2**. For more details about recording user scenarios, you can refer to **How Tos->Creating Transaction** section in this guide.

Step 3: Parameterize Transaction

Once you have your basic recorded transaction, you have to parameterize the transaction to vary the input to the server. This will help you to emulate real-world testing and avoid errors arising out of duplicate values.
Identify the list of values to be parameterized. Replace the recorded values with parameters to pass different set of data to the server for each virtual user who perform the particular transaction. For example, user name and password values used to login into the application can be parameterized using the Dataset option in a login transaction to pass different user name and password for each virtual user. The values for the parameter can be fetched from an external datasource (CSV or Excel or database) or from a cookie or from a previous response body or from a hidden field or from a previous URL or by executing a Javascript or using a constant value. For more details about parameterization, you can refer to **How Tos>Parameterizing Transaction** section in this guide.

Step 4: Percentage user configuration for Transaction

After recording the transaction configure the number of users to perform this transaction by specifying % of users. During load testing playback the number of virtual users for this transaction will be calculated based on the total number of users configured for the load test.

For Example: Configure the transaction wise user split as below,

User Scenario	**Transaction Name**	**% of users**
Browsing the web application Home Page	HomePage	40 %
Browsing the product catalog	ProductCatalog	40 %
Buying some product from the web site	BuyProduct	20 %

The sum of percentage of users of the transactions in the load test should be equal to 100 %.

Step 5: Configure Type of Load and Virtual Users

In this step, you examine your web application's behavior under simulated load conditions. This will help you to identify whether your application is trending toward or away from its defined performance objectives. Once you identify the load type to be simulated based on the instructions given in **Step 4** in **Planning Your Load Test**, select the appropriate load type and configure the workload details. For example, let us consider the following load for a sample online bookstore application.

- **Load Type - Normal**
- **User Count - 100 users**
- **Number of Repetitions - 2**

In the above configuration, QEngine will execute each transaction two repetition.

Load Distribution for the above load configuration in **Step 4** will be as follows:

User Transactions	**% of Users**	**Load Distribution for each transactions from the total number of users configured**
Browsing the web application Home Page	40%	40 users
Browsing the product catalog	40%	40 users
Buying some product from the web site	20%	20 users

For more details about configuring load or number of users, you can refer to **How Tos->Configuring Load Type** section in this guide.

Step 6: Playback Load test

Playback the load test. You can playback the load test created in **Step 5** using the following steps:

- Click on the **Start Play** button available in the load test screen.
- This will start the initialization for the load test and will show the play status screen. In the play status screen, you can see the runtime metrics of the load testing such as Hits to the server, Errors occured, Active number of users etc.
- During load test execution, QEngine automatically simulates the test runs for various workload models emulating real-time user roles and access patterns. After test completion QEngine will generate reports for the execution and will show it in a popup window.

Step 7: Analyze Test Results

The final step is to analyze the test results which is both the most important and the most difficult part of performance testing process. To make this process easier, the reports and graphs page that gets displayed after load test execution is organized in a clear and easy to understand format. This will help you to just click the various links in the left pane and identify the performance bottlenecks.

When you run the load test for the test case created in **Step 4**, the results will be as follows:

- **Avg. Hits/sec or load generated against server**: 20 hits per second.
- **Avg. Page Download Time**: 4,371 milliseconds.
- **Avg. Throughput or Data Transfer Rate**: 83,552 bytes per second.
- **Avg. Response Time or How Fast your Server Responds**: 4,334 milliseconds response time.
- **Avg. Error Percentage**: 25% for 120 active users when the elapsed time is at 30 seconds.
- **Server Monitoring Results**:
    - % CPU Usage: 40 percent
    - Memory Usage: 60 percent

For more details about analyzing test results, you can refer to **How Tos->Analyzing Test Results** section in this guide.

Sample Application

A sample suite named **WebPerformanceDemo** has been bundled with QEngine in <QEngine_Home>/projects/WebPerformanceDemo directory to illustrate the various features of Web Performance Test tool. To understand the sample application and how to execute the sample scripts, please refer to **Help->Tutorials->Adding/Executing Web Performance Test Cases** topic by browsing to the http://qengine.wiki.zoho.com.

How Tos

Working with Test Suites

What is a Suite ?

Suite is a collection of test scripts, test data, environment data and configuration data.

Why should I create a new Suite ?

Suite helps you to store test scripts or test information in an organized manner and share configuration and environment data amongst the different test types it contains.

How do I create a new Suite ?

To create a new Suite, from the Suite Manager UI , click the Create Suite button available above the Available Suites table.

How do I import a Suite ?

Use the Import Suite option in the Suite Manager page to import a Suite. The Suite Manager page gets displayed after connecting to the server and after logging-in. From the Suite Manager page, click the Import Suite button available above the Available Suites table to browse and specify the .qed file and click on the Import.

How do I export a Suite ?

A Suite can be exported by clicking the icon next to each of the suite in the Suite Manager page. The Suite Manager gets displayed after connecting to the server and after logging-in. Choose the icon or the Export Suite option in the Quick Task section in the Suite Manager page to export the suite. The exported Suite will be available as a .qed file under < QEngineHome > directory.

Creating Load Test

How are Load Test Created ?

To create new load test, choose **New Load Test** link from Quick Links placed below the tabs or right click on the suite in the left tree and choose **New Load Test** option from the context menu. This will show the New Load Test div, there specify the name for the load test and click on the OK button. The load test will be created under the suite with empty configuration.

What is a Transaction ?

Transaction is an user scenario in the application which is accessed by the virtual users during load testing. Transaction will contain the sequence of URL's with associated parameters which is recorded from the browser. Transactions in QEngine will be present under <QEngine_Home>/projects/ <Suite_Name>/webperfcases/businesscase/<Transaction_Name>/ folder. It will containg <Transaction_Name>.bcs and <Transaction_Name>.xml files.

How are Transaction for the load test created ?

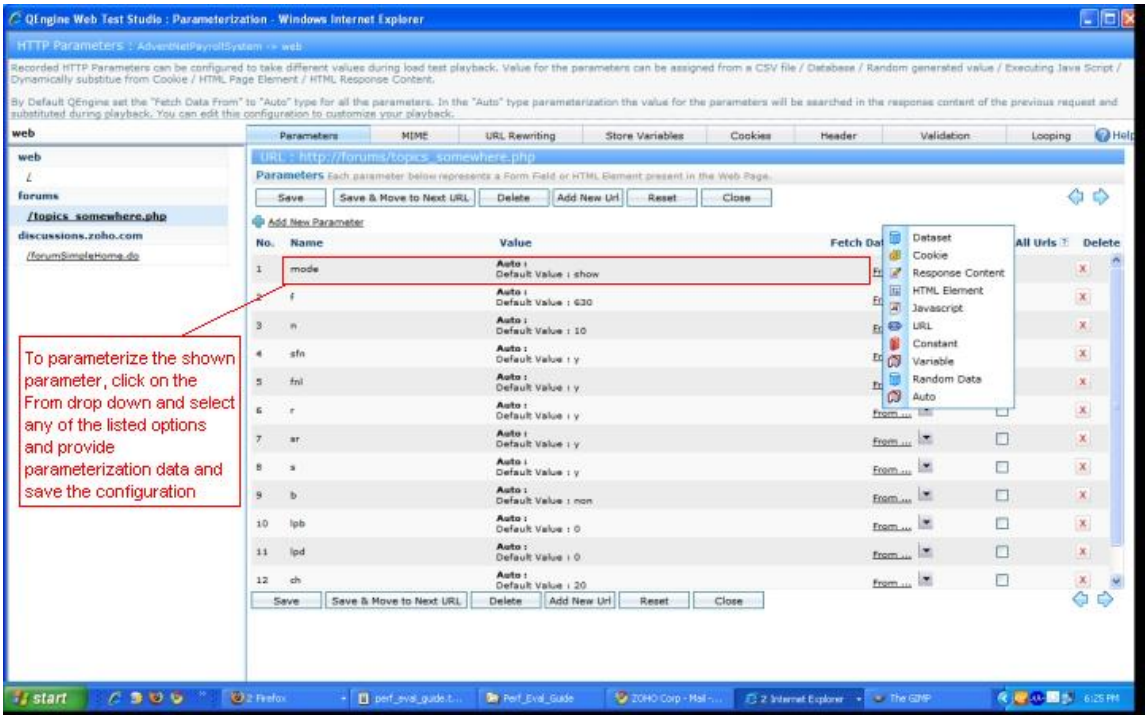
To create new transaction, choose **New Transaction** option from Load test screen shown on choosing the load test listed in the tree. This will show New Transaction div, there specify the name of the new transaction and click the OK button. This will start the recording to record the user scenario and will launch the new browser window. In the browser window launched from QEngine, load the application URL and browse the user scenario that needs to be accessed during the load testing. The URL loaded on the browser window will be automatically recorded and script will be generated. On completing the user scenario click on the **Stop Recording** button and save the transaction.

Parameterizing Transaction

How do I pass different data for form fields or query parameters ?

To pass different data for form fields or query parameters:

1. Click on the **Edit** link available below **HTTP Parameters / Validation** column for the each transaction from the transaction table.
2. This will display the parameterization screen for the with the recorded URL's listed in a tree.
3. On selecting any URL from the tree the associated **Parameter / Mime Parameter / Cookies** etc. will be listed in the right side of the screen.
4. Select the required URL from the tree and select the **Parameters** tab. This will display all the form fields or query parameters in the selected URL.
5. Select the drop down under **Fetch Data From** appropriate to the parameter you wish to parameterize. The UI will show the various options for parameterizing the selected parameter as shown below:



6. To pass different data, you can fetch the data from a cookie, from a previous response body, from a hidden field, from a previous URL, by executing Javascript or using a constant value. To

know the details of how to fetch values and parameterize URL parameters, please refer to the following URL:
http://qengine.wiki.zoho.com/About-Get-Post-Data-Parameterization.html#Handling_Data_Parameterization

How do I pass different data in HTML form fields for each virtual user ?

To pass different data in form fields for each virtual user:

1. Click on the **Edit** link available below **HTTP Parameters / Validation** column for the each transaction.
2. This will display the parameterization screen for the with the recorded URL's listed in a tree.
3. Select the required URL from the tree and select the **Parameters** tab. This will display all the form fields or query parameters in the selected URL.
4. Select the drop down under **Fetch Data From** appropriate to the parameter you wish to parameterize. The UI will show the various options for parameterizing the selected parameter as shown above.
5. To pass different data for each virtual user, select the Dataset option. This displays the View Data Configuration screen from where you can select existing data source or add new datasource to fetch values from a CSV file or a Database or Excel file for each virtual user.
   To know the details of how to fetch values from a Dataset for each virtual user, please refer to the following URL:
   http://qengine.wiki.zoho.com/Parameterizing-Get-Post-Data-Using-Dataset.html

How to parameterize some dynamic parameters which i do not provide during recording ?

If the parameter for the URL request is taken from any hidden element present in the web page, then those parameters will be automatically parsed and taken by the QEngine. User does not require to parameterize such parameters. Thus the user is only need to configure the parameter which are provided by him during recording.

Can I reuse the same script with different data for iterative execution of a business process ?

Yes, you can use the transaction with different data for iterative execution of a business process using the Dataset option. This will dynamically fetch the values from a CSV file or a database or Excel file and replaces it with the configured form fields or query parameters for each execution of a business process.

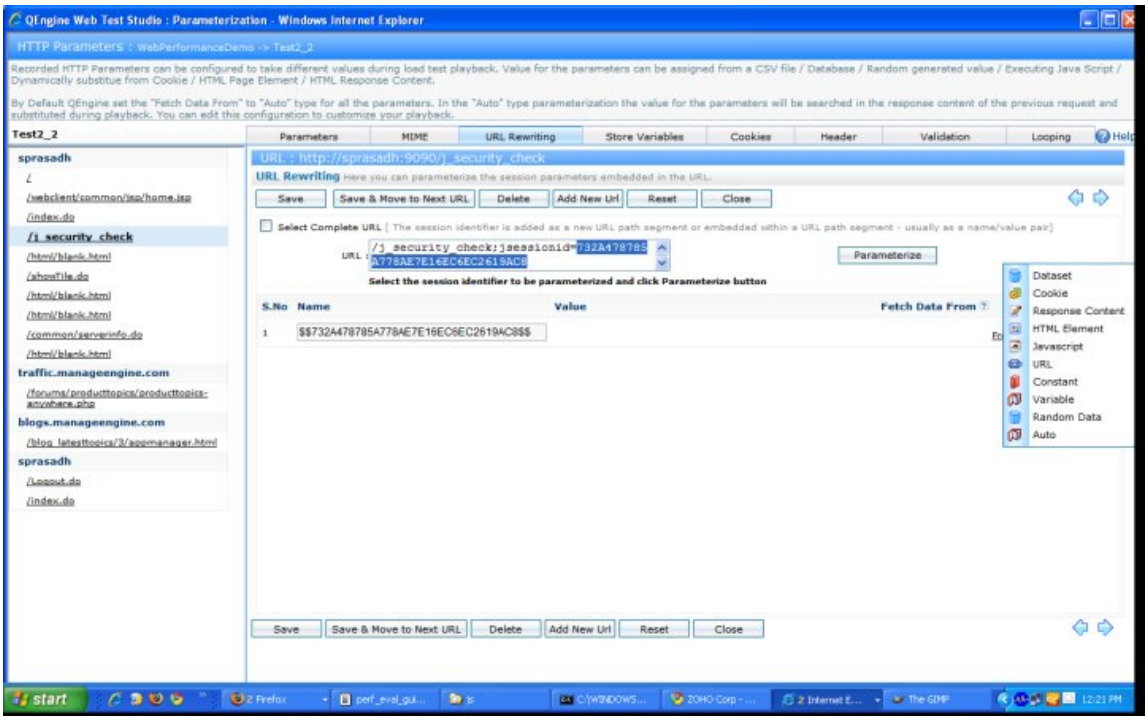
What is url-rewriting ?

URL-Rewriting inserts the session identifier or session key as a parameter in every HTML link in a web page. URL-Rewriting is used by the server when the browser has cookie support disabled.

How do I use url-rewriting for session tracking ?

To track session variables or tokens in recorded URLs,

1. Click on the **Edit** link available below **HTTP Parameters / Validation** column for the each transaction.
2. This will display the parameterization screen for the with the recorded URL's listed in a tree.
3. Select the required URL from the tree and select the **URL Rewriting** tab.This will display the URL in the URL field.
4. From URL field, highlight the session identifier to be parameterized and click on **Parameterize** button. This will display the selected session identifier in Parameter Name field.
5. From the Parameterize column, click on the list button under **Fetch Data From** column, which will list the options to parameterize as shown in the image below:



6. You can fetch the value for the session identifier from a hidden field or cookie or previous response body or previous URL or by executing Javascript or constant value. To know the details of how to handle session identifiers, please refer to the following URL: http://qengine.wiki.zoho.com/Parameterizing-Get-Post-Data-Using-Dataset.html

Creating User Transactions

Can I create multiple user scenarios for a load test?

Yes, you can create multiple user scenarios for a load test and associate percentage of users appropriately.

To create Multiple Transaction for a load test follow the below steps:

1. Choose the Load Test from the Suite tree.
2. Click on **New Transaction** or **Import Transaction** to add a transaction to the load test.
3. On clicking **New Transaction** provide name for the transaction and record the scenario.
4. On clicking **Import Transaction**, choose the transaction to be imported from the available transactions and provide name in which the transaction should be imported inside the load test.
5. Similarly after adding the transaction follow the Step 2 to Step 4 to add multiple transaction to the load test.
6. To know more on adding transactions follow the below help link: http://qengine.wiki.zoho.com/Configuring-Transactions.html

Can I change my server host without re-recording URLs ?

Yes, you can change your server host / port without re-recording the URLs.

1. Click on the Settings link present below the suite name in the Suite Tree.
2. Under Load Test Settings, choose the check box for **Test application on alternate host**
3. Provide value for alternate **Host** and **Port** and click on Apply and save the settings. To know more on this follow : http://qengine.wiki.zoho.com/Setting-Runtime-Option.html#load_test

Can I vary the user load for each user scenario ?

Yes, you can vary the user load for each user scenario.

1. Choose the load test from the Suite Tree.
2. From the available tranasction table configure the % Load for the transaction.
3. The sum of the percentage of all transaction in the load test should equal to 100.

What is think time ?

Think time is the time that a real user takes while filling any forms or clicking any link etc. in the web page before submitting the details to the server.

What is repeat delay ?

A virtual user is assigned to play back a single transaction repeatedly. The repeat delay setting is the number of milliseconds to delay between repeats.

Configuring Load Test

Can a load test be ramped up and ramped down ?

Yes, you can create load test with ramp up and ramp down load models. Ramp up load test simulates heavy load by gradually increasing the number of users at defined periods until the count reaches the maximum number of users. This will help you to determine the peak load
at which your web site fails to respond. Ramp down load test gradually drops off load at defined periods from a peak value to a lowest value. You can create load test with ramp-up and ramp-down to compare and determine how well your web site responds at peak hours of the system and when it goes back to an idle state.

1. Choose the Load Test from the suites tree.
2. From the Load Test screen choose **Ramp-up** or **Ramp-down** load type under **Load Details**.
3. Configure the appropriate values for Startwith, Increment/Decrement and Limit to fields. For more details, refer to the document in the following URL: http://qengine.wiki.zoho.com/Adding-Load-Details.html#ramup_mode

How do I create a goal-based load test ?

Goal-based load testing enables you to determine how much load your system can support before reaching a limiting factor such as exit the load test when the CPU reaches 80% or when the server crashes. etc. To create goal-based load test,

1. Choose the Load Test from the suites tree.
2. From the Load Test Screen choose **Burn-in** load type under **Load Details .**
3. Configure the appropriate value for UserCount.
4. Select the appropriate exit criteria from the Exit Criteria Configuration pane. This is mandatory for goal-based testing. For more details, refer to the document in the following URL: http://qengine.wiki.zoho.com/Configuring-Exit-Criteria.html

Can I vary the connection speed for virtual clients ?

Yes, you can vary the connection speed with which the virtual client or user connects to the web application server using the Modem Simulation option in the Load test screen.

1. Choose the Load Test from the tree.
2. Click on the **Modem Simulation** link from available under the Load Details section.

3. This will show the Modem Simulation popup which will show the various connection speed in the **Connection Speed** combo.
4. By default **None** will be selected which uses the default Connection speed available in the network.
5. From the Modem Simulation combo, select any of the options such as Analog modem (14.4 kbps, 28.8 kbps, and 56 kbps), ISDN modem (64 kbps), Dual ISDN modem (128 kbps), or DSL modem (256 Kpbps, 512 kbps).
6. Click on **Save** to save the configuration in the load test.

How do I collect data from a remote server in a load test ?

To collect data from a remote server:

1. Choose the load test from the suite's tree.
2. From the load test screen, click on the **Server Monitoring** link to add multiple server monitors. To know the details, please refer to the following URL: http://qengine.wiki.zoho.com/Server-Monitoring.html

How do I find database related issues in a load test ?

To find database related issues in a load test, you can configure Database monitors for MySQL, Oracle and MSSQL database and analyze the database monitoring graphs after load test execution.

1. Choose the load test from the suite's tree.
2. From the load test screen, click on the **Database Monitoring** link to add multiple server monitors. To know the details, please refer to the following URL: http://qengine.wiki.zoho.com/Database-Monitoring.html

What are the common issues faced in server monitoring ?

Some of the common issues faced in server monitoring are due to the following reasons:

1. The remote host and the client machine might not be in the same domain.
2. Check whether WMI service is running or not in the remote and local machines.
3. Check whether RPC server is running or not in the remote and local machines.
4. Check if firewall is not enabled in the remote machine.

Adding Validation

Can I add validation to my load test ?

Yes, you can add validation in load test transactions using the **Validation** tab in the Parameterization window. To know the details, please refer to the following URL:

http://qengine.wiki.zoho.com/Response-Validator.html

Running Your Load Test

Are there any runtime settings to be done ?

Yes, there are a wide variety of runtime options available. All are optional. For details, refer to the following URL:

http://qengine.wiki.zoho.com/Setting-Runtime-Option.html

How do I run my load test case with the configured load ?

To run your load test with the configured load,

- Choose the load test to be run from the suite tree.
- Click on **Start Play** button to run the load test. Displays runtime graphs during execution and after test completion, displays the detailed reports and graphs.

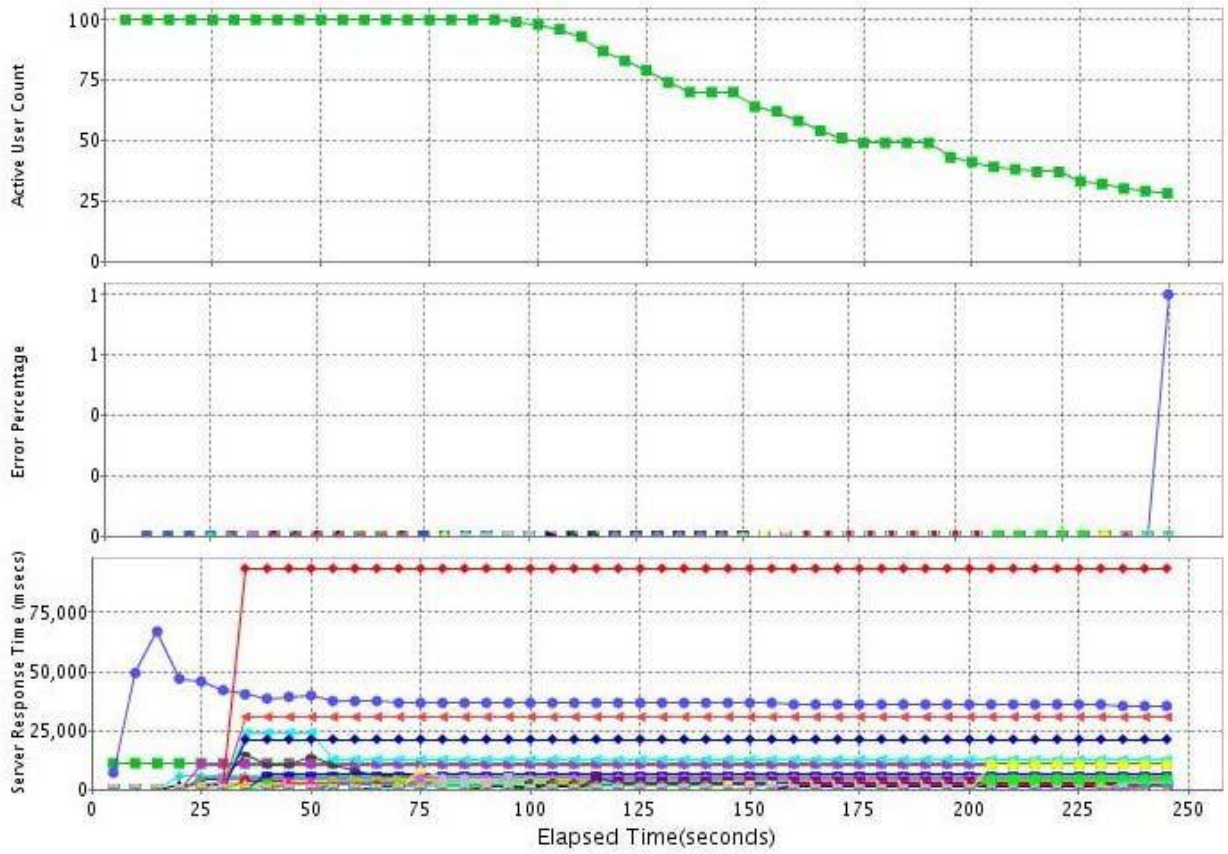
Can I view dynamic graphs during load test execution ?

Yes, you can view dynamic graphs during load test execution.

Analyzing Test Results

How do I identify the performance bottlenecks ?

To identify performance bottlenecks, after load test execution, from the Reports , click on the link **Bottleneck analysis** link from the left pane. This will display the graph as shown in the sample screen below:



This graph provides a consolidated graph view to identify the performance bottlenecks in your web application. Plots the Elapsed time (in seconds) in the X-axis and the Server Response Time, Error Percentage and the Active User Count in the Y-axis. At one shot, you can identify the server response time for individual URLs, error percentage and the number of active users over elapsed time.

This will identify you whether Error Percentage or Response time increases with increase in active user count.

How do I determine the user load generated against my web server ?

To determine the load generated against your web server, after load test execution, from the Reports, click on the link **Load Reports**. This displays the

Time Vs User Count and **Time Vs Hits per second** graphs for each business case and consolidated report for all the business.

1. **Time Vs User Count** - This graph shows the user addition interval over elapsed time for each transaction and consolidated user count report for all the transactions.
2. **Time Vs Hits per second** - This graph shows the Elapsed time vs Number of requests handled successfully, per unit time for each transaction and consolidated hits per second report for all the transactions.

How do I determine the data transfer rate ?

To determine the data transfer rate, after load test execution, from the Reports, click on the link **Load Reports** link. This displays the **Time Vs Throughput** Graph for each transaction and consolidated throughput report for all the transactions. The mean value for the throughput through out the load testing will be shown below. Throughput is amount of data transferred between the client and server in a given period of time.

How do I determine the server response time ?

Response time is time between request sent to the server and first byte of response received from the server.

To determine the server response time, after load test execution, from the Reports, click on the link **Response Time Reports** link. This displays following graphs:

1. **Mean response time for each URL in the transaction** - Use this to determine which URL takes longer time for processing in the server.
2. **Mean response time for each transaction** - Use this to determine which transaction takes longer time to complete by the virtual user.
3. **Time Vs Response Time Vs Active User Conunt(Transaction wise)** - Use this to determine how response time for any given transaction changes with respect to increase in active number of users.
4. **Time Vs Response Time Vs Active User Count (All Transactions)** - Use this to determine how response time changes with respect to increase in active number of users.

How do I determine the page download time ?

Page download time is time taken between First byte of response received to last byte of response received.

To determine the page download time, after load test execution, from the Reports, click on the link **Page Reports**. This display the following graphs:

1. **Time Vs Page Download Time** - Use this to determine peak download time when the server is loaded.
2. **Time Vs Page Download Time -** Use this to determine the peak download time in any specific transaction when the server is loaded.
3. **Page Size Report** - Use this to determine large files that is being downloaded from the server.
4. **Page Download Report** - Use this to determine the pages that take longer time to download. Also check the associated images download time separately.

How do I determine the errors in my web page ?

To determine the errors in your web page, after load test execution, from the Reports, click on the link **Error Reports**. This display the following graphs:

1. **Error Distribution** -Use this to determin Transaction wise percentage errors.
2. **Time Vs Error % Vs Active User Count** - Use this to determine whether Percentage of Error increases with Active User Count.
3. **Time Vs Error % Vs Active User Count (Transaction wise)** - Use this to determine whether Percentage of Error in a transaction increases with Active User Count. So that you can identify the part of the application which is causing the error.

How do I determine the CPU and memory usage ?

If you have configured and selected server monitors for the load test, then after test execution, from the Reports, click on the **Server Monitor Reports** link. This displays following reports :

1. **CPU Usage Report** - Use this to determine the spike in CPU usage during load testing. For each monitor it shows separate line.
2. **Memory Usage Report** - Use this to determine the memory usage by the server during load testing.

How do I identify the database issues ?

If you have configured database monitors for the load test, then after test execution, from the Reports, click on the **Database Monitor Reports** link. This displays following reports:

1. **Transaction Summary** - This graph shows the total requests, total bytes sent and total bytes received.
2. **Connection Summary** - This graph shows the number of open connections and the total number of aborted connections.

Additional Features

How do I run load test from Command Line ?

To know the details of how to run load test cases from command line, please refer to the following URL:

http://qengine.wiki.zoho.com/Regression-Testing.html

Can I change the order of load test execution ?

Yes, you can change the order of load test execution using the automatic or manual sequencing option.

To know the details of test sequencing, please refer to the following URL:

http://qengine.wiki.zoho.com/Sequencing-Test-Scripts-1.html

Performance Tips & Tricks

Following are the do's and dont's to consider while performing the load test:

Do's

- Identify the appropriate system requirements to mirror your production environment.
- Consider using think time to pause between requests based on the operations performed in the web application to capture real-life testing. By default, think time is not considered during load test execution. You can configure the appropriate think time for each **submitURL()** statement in the transaction and choose the **Use think time** option in **Load Test settings**.
- While configuring the load to ramp up or ramp down users for a load test, consider including a buffer time between the incremental increases of users.
- Parameterize your load test transaction after recording, to send different set of data for each virtual users or to avoid errors due to duplication of values.
- Identify and prioritize the key user scenarios according to critical functionality and time-consuming transactions.

Dont's

- Do not place too much load on a single client machine. This will create bottlenecks during load testing. To avoid this, use distributed load testing to distribute the load on multiple client machines.
- Do not run your load tests in a live production environment. Use an in-house test environment that mirrors your production environment.
- Do not allow the CPU and memory usage of your client machines that generates load to cross the threshold limit. Since, this might not give appropriate data in the load test reports.