Load Runner 9.0
Training Courseware

By

Softsmith Infotech
www.softsmithinfotech.com

Table of Contents

Load Test:

Load Tests are end to end performance tests under anticipated production load. The objective of such tests are to determine the response times for various time critical transactions and business processes and ensure that they are within documented expectations (or Service Level Agreements - SLAs). Load tests also measures the capability of an application to function correctly under load, by measuring transaction pass/fail/error rates.

Load Tests are major tests, requiring substantial input from the business, so that anticipated activity can be accurately simulated in a test environment. If the project has a pilot in production then logs from the pilot can be used to generate 'usage profiles' that can be used as part of the testing process, and can even be used to 'drive' large portions of the Load Test.

Load testing must be executed on "today's" production size database, and optionally with a "projected" database. If some database tables will be much larger in some months time, then Load testing should also be conducted against a projected database. It is important that such tests are repeatable, and give the same results for identical runs. They may need to be executed several times in the first year of wide scale deployment, to ensure that new releases and changes in database size do not push response times beyond prescribed SLAs.

What is the purpose of a Load Test?

The purpose of any load test should be clearly understood and documented. A load test usually fits into one of the following categories:

Quantification of risk. - Determine, through formal testing, the likelihood that system performance will meet the formal stated performance expectations of stakeholders, such as response time requirements under given levels of load. This is a traditional Quality Assurance (QA) type test. Note that load testing does not mitigate risk directly, but through identification and quantification of risk, presents tuning opportunities and an impetus for remediation that will mitigate risk.

Determination of minimum configuration. - Determine, through formal testing, the minimum configuration that will allow the system to meet the formal stated performance expectations of stakeholders - so that extraneous hardware, software and the associated cost of ownership can be minimized. This is a Business Technology Optimization (BTO) type test.

Assessing release readiness by: Enabling you to predict or estimate the performance characteristics of an application in production and evaluate whether or not to address performance concerns based on those predictions. These predictions are also valuable to the stakeholders who make decisions about whether an application is ready for release or capable of handling future growth, or whether it requires a performance improvement/hardware upgrade prior to release.

What functions or business processes should be load tested?

Basis for inclusion in Load Test	Comment
High frequency transactions	The most frequently used transactions have the potential to impact the performance of all of the other transactions if they are not efficient.
Mission Critical transactions	The more important transactions that facilitate the core objectives of the system should be included, as failure under load of these transactions has, by definition, the greatest impact.
Read Transactions	At least one READ ONLY transaction should be included, so that performance of such transactions can be differentiated from other more complex transactions.
Update Transactions	At least one update transaction should be included so that performance of such transactions can be differentiated from other transactions.

Load Testing Process

- Identify the performance-critical scenarios.
- Identify the workload profile for distributing the entire load among the key scenarios.
- Identify the metrics that you want to collect in order to verify them against your Performance objectives.
- Design tests to simulate the load.
- Use tools to implement the load according to the designed tests, and capture the metrics.
- Analyze the metric data captured during the tests.

Load Test Terminologies:

Scenarios are anticipated user paths that generally incorporate multiple application activities. Key scenarios are those for which you have specific performance goals, those considered to be high-risk, those that are most commonly used, or those with a significant performance impact. The basic steps for identifying key scenarios are.

Metrics are measurements obtained by running performance tests as expressed on a commonly understood scale. Some metrics commonly obtained through performance tests include processor utilization over time and memory usage by load.

Response time is a measure of how responsive an application or subsystem is to a client request.

Throughput is the number of units of work that can be handled per unit of time; for instance, requests per second, calls per day, hits per second, reports per year, etc.

Workload is the stimulus applied to a system, application, or component to simulate a usage pattern, in regard to concurrency and/or data inputs. The workload includes the total number of users, concurrent active users, data volumes, and transaction volumes, along with the transaction mix. For performance modeling, you associate a workload with an individual scenario.

Resource utilization is the cost of the project in terms of system resources. The primary resources are processor, memory, disk I/O, and network I/O.

Scalability refers to an application's ability to handle additional workload, without adversely affecting performance, by adding resources such as processor, memory, and storage capacity.

Performance requirements are those criteria that are absolutely non-negotiable due to contractual obligations, service level agreements (SLAs), or fixed business needs. Any performance criterion that will not unquestionably lead to a decision to delay a release until the criterion passes is not absolutely required — and therefore, not a requirement.

Performance-testing effort in your project context should include:

- **Project vision.** Before beginning performance testing, ensure that you understand the current project vision. The project vision is the foundation for determining what performance testing is necessary and valuable. Revisit the vision regularly, as it has the potential to change as well.
- **Purpose of the system.** Understand the purpose of the application or system you are testing. This will help you identify the highest-priority performance characteristics on which you should focus your testing. You will need to know the system's intent, the actual hardware and software architecture deployed, and the characteristics of the typical end user.
- **Customer or user expectations.** Keep customer or user expectations in mind when planning performance testing. Remember that customer or user satisfaction

is based on expectations, not simply compliance with explicitly stated requirements.

- **Business drivers.** Understand the business drivers – such as business needs or opportunities – that are constrained to some degree by budget, schedule, and/or resources. It is important to meet your business requirements on time and within the available budget.
- **Reasons for testing performance.** Understand the reasons for conducting performance testing very early in the project. Failing to do so might lead to ineffective performance testing. These reasons often go beyond a list of performance acceptance criteria and are bound to change or shift priority as the project progresses, so revisit them regularly as you and your team learn more about the application, its performance, and the customer or user.
- **Value that performance testing brings to the project.** Understand the value that performance testing is expected to bring to the project by translating the project- and business-level objectives into specific, identifiable, and manageable performance testing activities. Coordinate and prioritize these activities to determine which performance testing activities are likely to add value.
- **Project management and staffing.** Understand the team's organization, operation, and communication techniques in order to conduct performance testing effectively.
- **Process.** Understand your team's process and interpret how that process applies to performance testing. If the team's process documentation does not address performance testing directly, extrapolate the document to include performance testing to the best of your ability, and then get the revised document approved by the project manager and/or process engineer.
- **Compliance criteria.** Understand the regulatory requirements related to your project. Obtain compliance documents to ensure that you have the specific language and context of any statement related to testing, as this information is critical to determining compliance tests and ensuring a compliant product. Also understand that the nature of performance testing makes it virtually impossible to follow the same processes that have been developed for functional testing.

- **Project schedule.** Be aware of the project start and end dates, the hardware and environment availability dates, the flow of builds and releases, and any checkpoints and milestones in the project schedule.

Load Runner:

HP LoadRunner, a tool for performance testing, stresses your entire application to isolate and identify potential client, network, and server bottlenecks.

HP LoadRunner load tests your application by emulating an environment in which multiple users work concurrently. While the application is under load, LoadRunner accurately measures, monitors, and analyzes a system's performance and functionality.

How LoadRunner Addresses the Performance Testing:

- LoadRunner reduces personnel requirements by replacing human users with virtual users or Vusers. These Vusers emulate the behavior of real users—operating real applications.
- Because numerous Vusers can run on a single computer, LoadRunner reduces the amount of hardware required for testing.
- The HP LoadRunner Controller allows you to easily and effectively control all the Vusers—from a single point of control.
- LoadRunner monitors the application performance online, enabling you to fine-tune your system during test execution.
- LoadRunner automatically records the performance of the application during a test. You can choose from a wide variety of graphs and reports to view the performance data.
- LoadRunner checks where performance delays occur: network or client delays, CPU performance, I/O delays, database locking, or other issues at the database server. LoadRunner monitors the network and server resources to help you improve performance.
- Because LoadRunner tests are fully automated, you can easily repeat them as often as you need.

Various Components of LoadRunner:

Vuser Generator:

Vuser Generator is the Script generation component of LoadRunner. This component has two main things and are described below:

Vusers In the scenario, LoadRunner replaces human users with virtual users or Vusers. When you run a scenario, Vusers emulate the actions of human users working with your application. While a workstation accommodates only a single human user, many Vusers can run concurrently on a single workstation. In fact, a scenario can contain tens, hundreds, or even thousands of Vusers.

Vuser Scripts. The actions that a Vuser performs during the scenario are described in Vuser script. When you run a scenario, each Vuser executes a Vuser script. The Vuser scripts include functions that measure and record the performance of your application's components.

Controller. You use the HP LoadRunner Controller to manage and maintain your scenarios. Using the Controller, you control all the Vusers in a scenario from a single workstation.

Load Generator. When you execute a scenario, the Controller distributes each Vuser in the scenario to a load generator. The load generator is the machine that executes the Vuser script, enabling the Vuser to emulate the actions of a human user.

Performance analysis. Vuser scripts include functions that measure and record system performance during load-testing sessions. During a scenario run, you can monitor the network and server resources. Following a scenario run, you can view performance analysis data in reports and graphs.

Working with LoadRunner

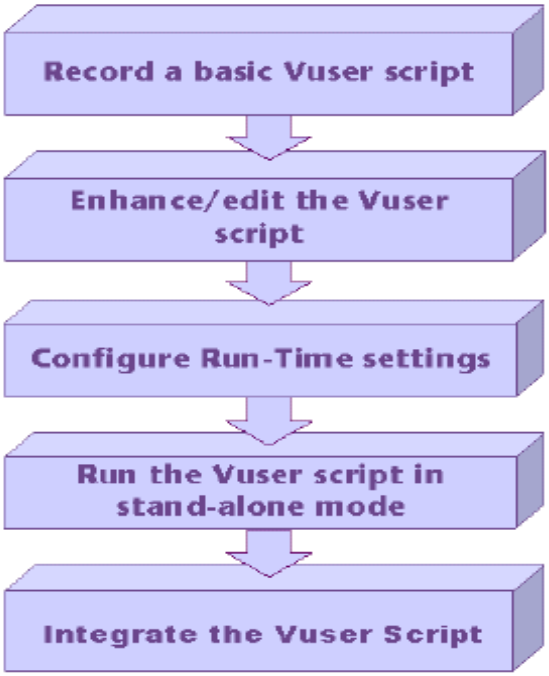
When testing or monitoring an environment, you need to emulate the true behavior of users on your system. HP testing tools emulate an environment in which users concurrently work on, or access your system. To do this emulation, the human was replaced with a virtual user, or a Vuser. The actions that a Vuser performs are described in a Vuser script. The primary tool for creating Vuser scripts is the Virtual User Generator, **VuGen.**

Vusers emulate the actions of human users by performing typical business processes in your application. The actions that a Vuser performs during the recording session are described in a Vuser script.

Using VuGen, you can run scripts as standalone tests. Running scripts from VuGen is useful for debugging as it enables you to see how a Vuser will behave and which enhancements need to be made.

Steps for Creating Scripts:

VuGen enables you to record a variety of Vuser types, each suited to a particular load testing environment or topology. When you open a new test, VuGen displays a complete list of the supported protocols.



This window opens up as soon as you open the VuGen. You can select the protocol of you application and click ok. For most of the web application its Web (HTTP/HTML) Protocol

Open VuGen

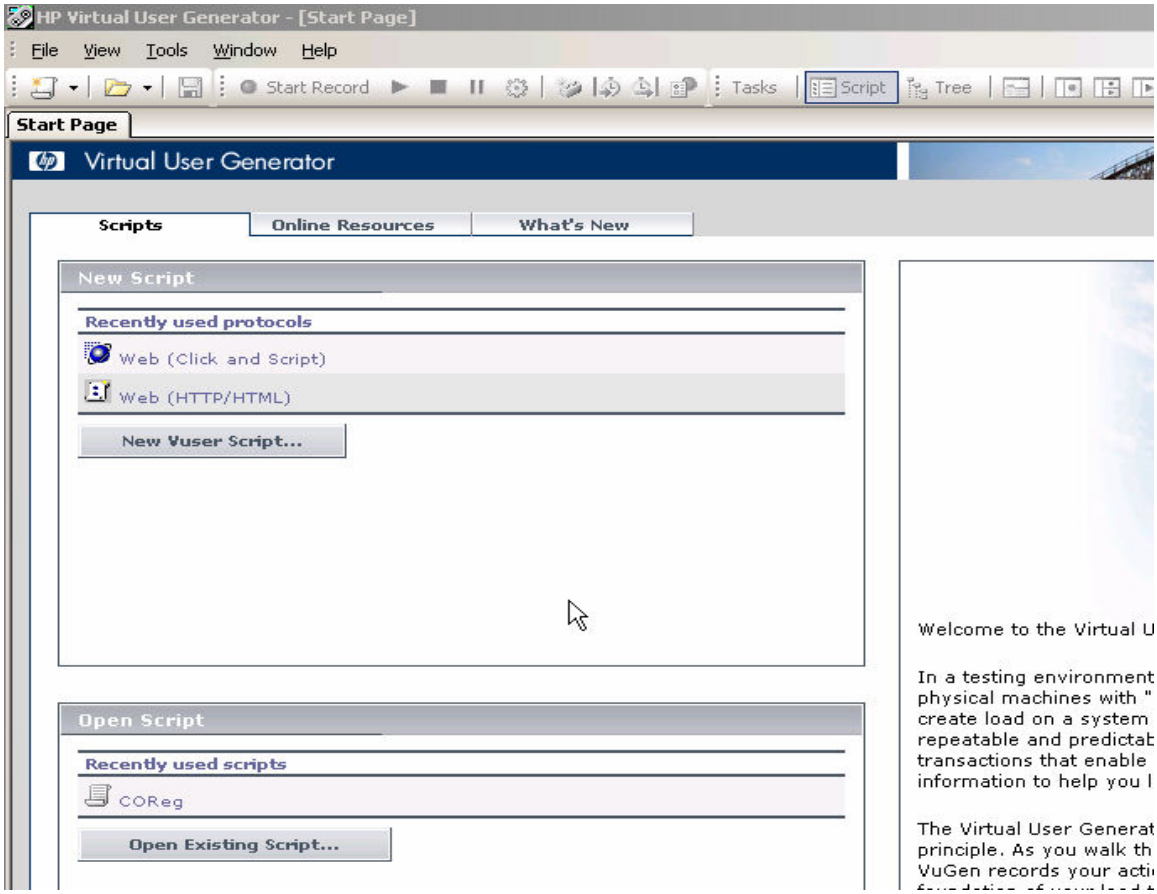
To start VuGen, choose Start > Programs > <*App_Name*> *(*for example LoadRunner*)* > Applications > Virtual User Generator from the Start menu.

- To open an existing script, not in the recent list, click Open Existing Script.
- To create a new script using a recent protocol, click the protocol in the

Recently used protocols list.

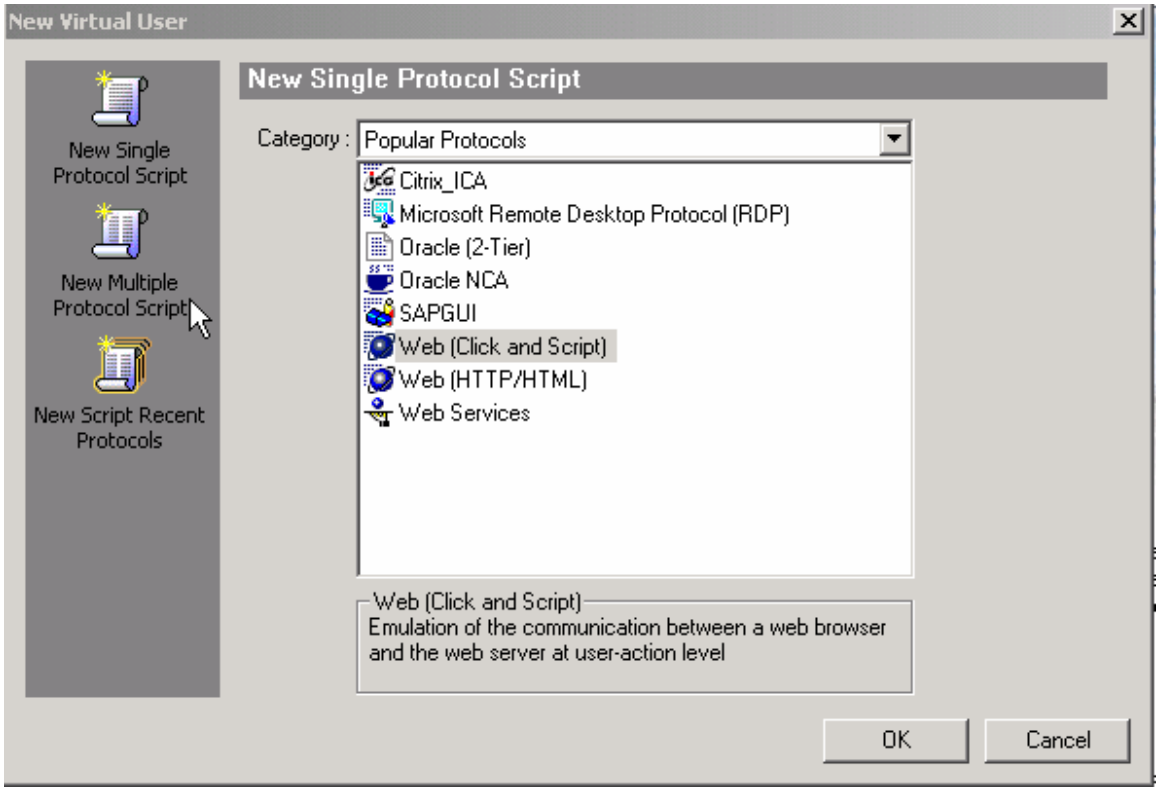
To create a new script in a protocol that is not listed, click New Vuser Script.

Choose File > Zip Operations > Import From Zip File … to open an existing cript from a zip archive.



Now click on the New Protocol Script and you will see the following Window. From this window you have to choose the protocol on which the application you are going to load test works.

VuGen provides a variety of Vuser technologies that allow you to emulate your system. Each technology is suited to a particular architecture and results in a specific type of Vuser script. For example, you use Web Vuser Scripts to emulate users operating Web browsers. You use FTP Vusers to emulate an FTP session. The various Vuser technologies can be used alone or together, to create effective load tests or Business Process Monitor profiles.

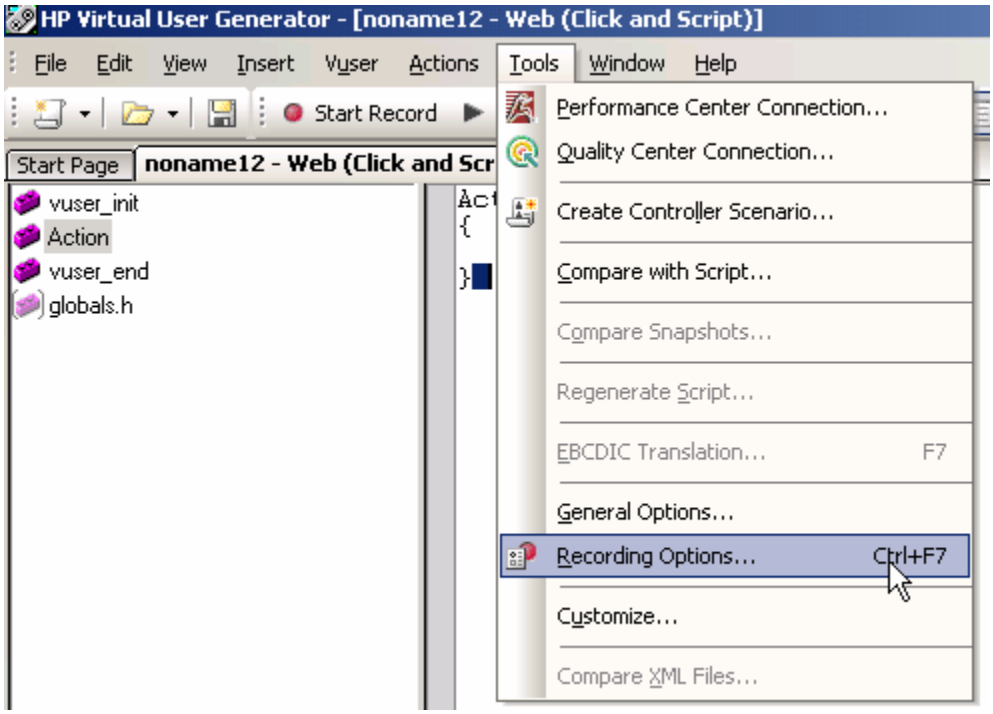


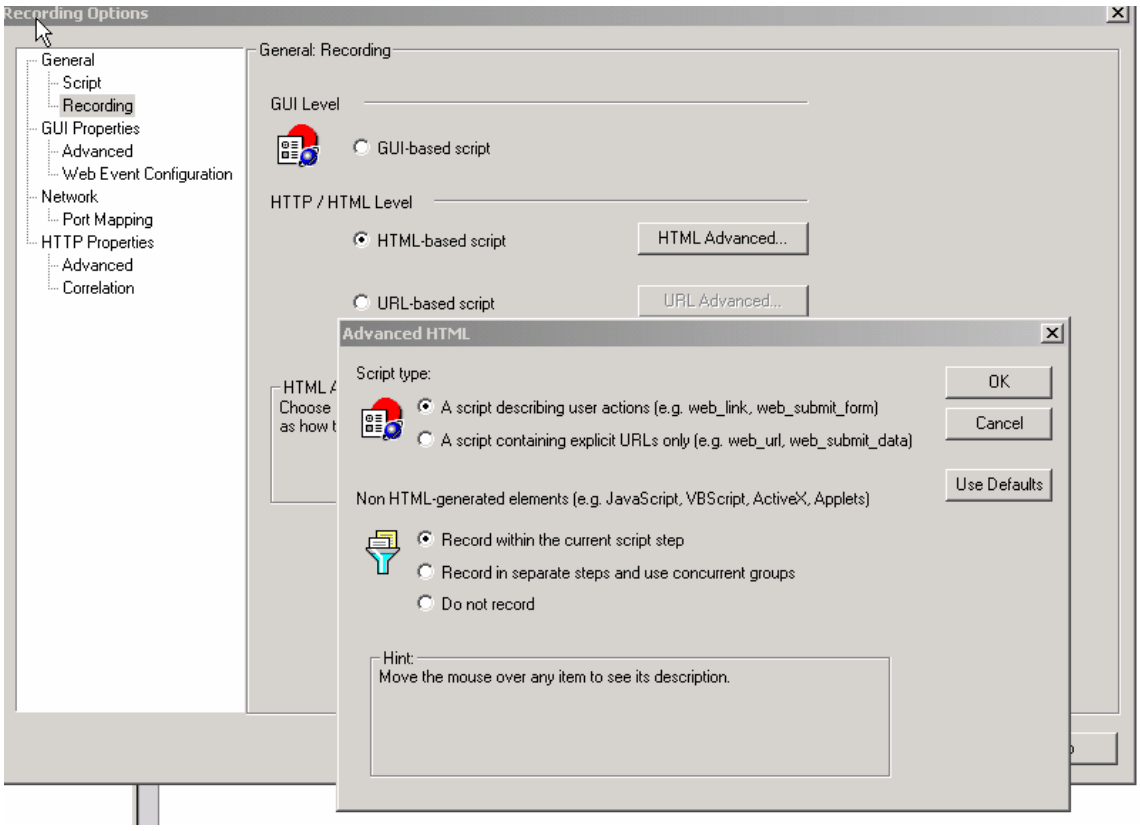
Now set the General options for VuGen.

For Example To set the environment-related options:

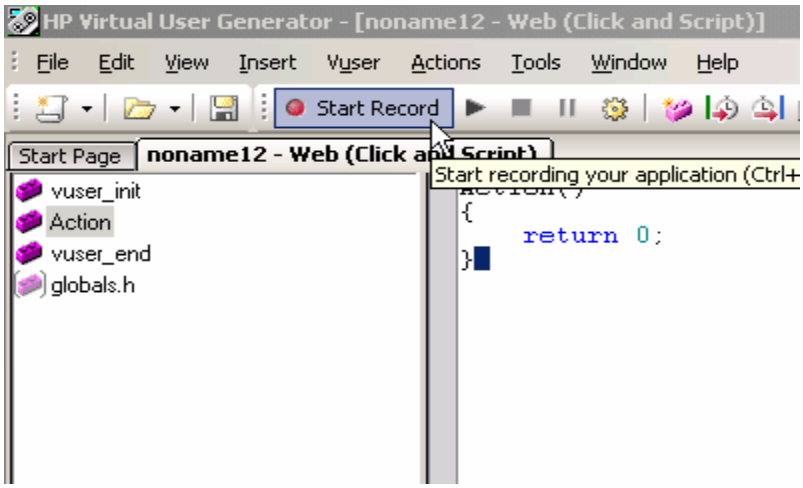
- Select Tools > General Options and click the Environment tab.
- To save the current script information for auto recovery, select the Save AutoRecover Information option and specify the time in minutes between the saves.
- To set the editor font, click Select Font. The Font dialog box opens. Select the desired font, style, and size and click OK. Note that only fixed size fonts (Courier, Lucida Console, FixedSys, and so on) are available.
- To use a comparison tool other than WDiff, select Use custom comparison tool and then specify or browse for the desired tool.
- Click OK to accept the settings and close the General Options dialog box.

Now Set the Recording Options for recording the user actions of the application under load test.

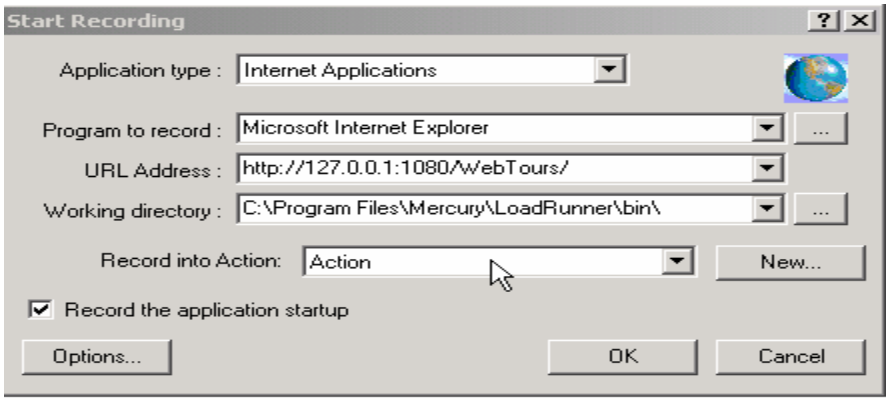




Now that you are ready for recording Click on the record button



Give the Url of the application that needs to be load tested like the one below



The following table describes the criteria for determining the business functions or processes to be included while recording

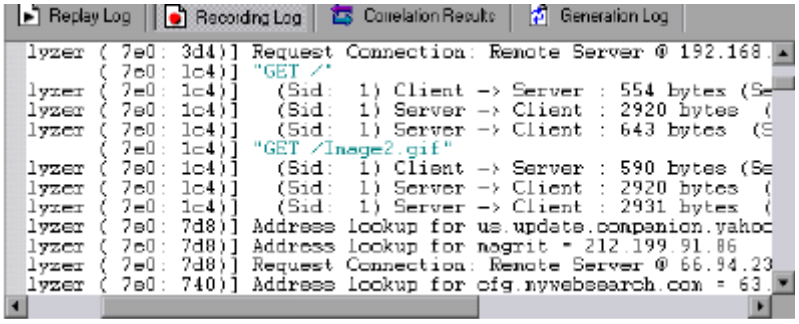
Basis for inclusion in Load Test	Comment
High frequency transactions	The most frequently used transactions have the potential to impact the performance of all of the other transactions if they are not efficient.
Mission Critical transactions	The more important transactions that facilitate the core objectives of the system should be included, as failure under load of these transactions has, by definition, the greatest impact.
Read Transactions	At least one READ ONLY transaction should be included, so that performance of such transactions can be differentiated from other more complex transactions.
Update Transactions	At least one update transaction should be included so that

	performance of such transactions can be differentiated from other transactions.

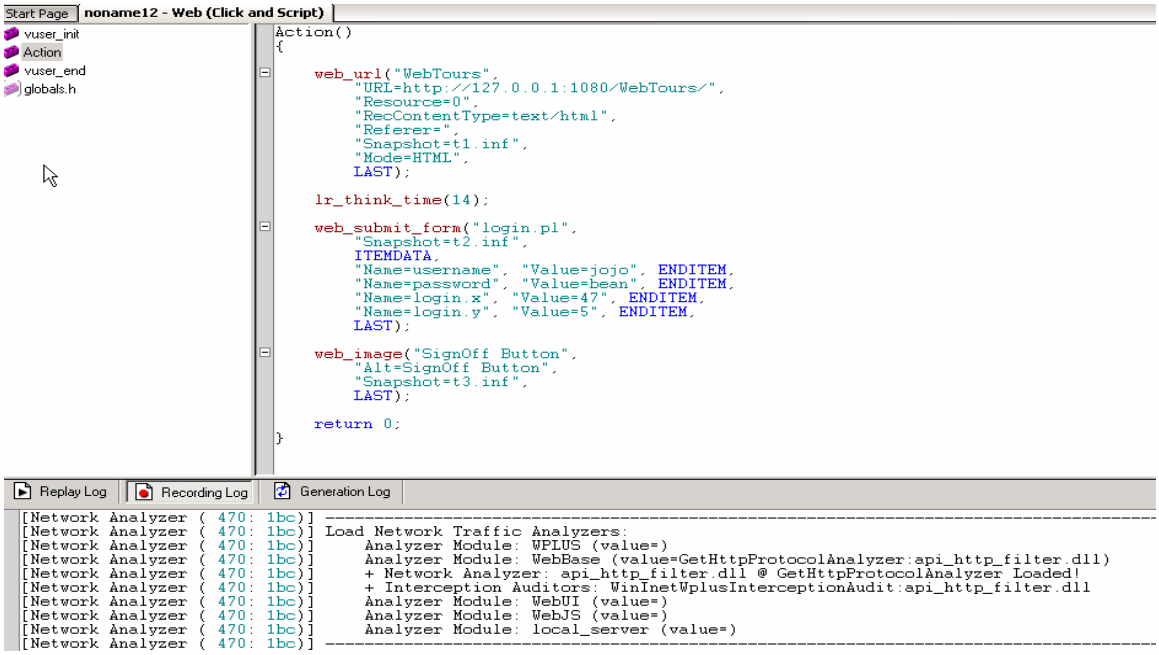
Now record the transaction in either Vuser_init or Vuser_end or action by using the recording tool bar.

Once the recording is over a recording log is also generated

To view a log of the messages that were issued during recording, click the Recording Log tab. You can set the level of detail for this log in the advanced tab of the Recording options.

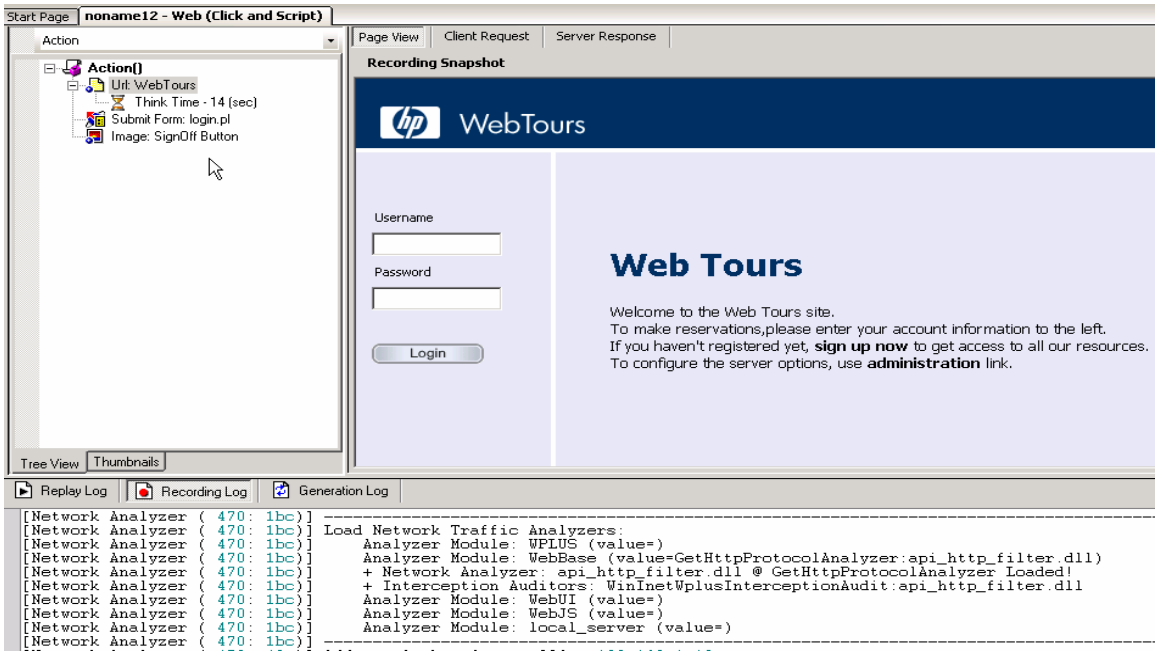


Now the script will be generated for the recorded user actions and will be displayed like this



The above one is called the Script view. It can also be viewed as Tree view

Tree view:

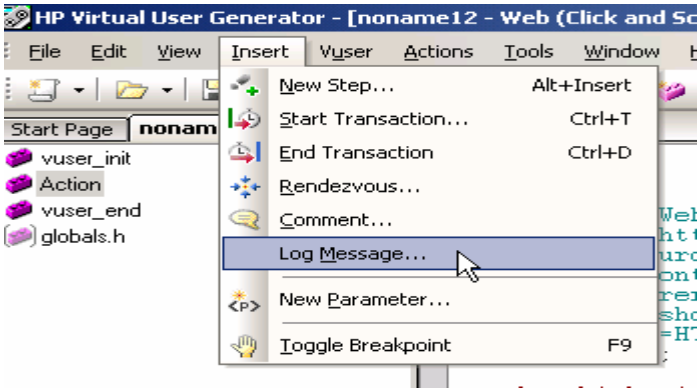


The Next step is enhance the script

This includes

1) Inserting Transaction statements
2) Log messages
3) Comments
4) Inserting Functions
5) Performing Correlation
6) Parameterize
7) Inserting Rendezvous Points
8) Doing some run time settings.

To Insert Inserting Transaction statements Log messages, Comments, Inserting Functions



Now you need to perform Correlation in the script. The purpose of Correlation is

To simplify or optimize your code

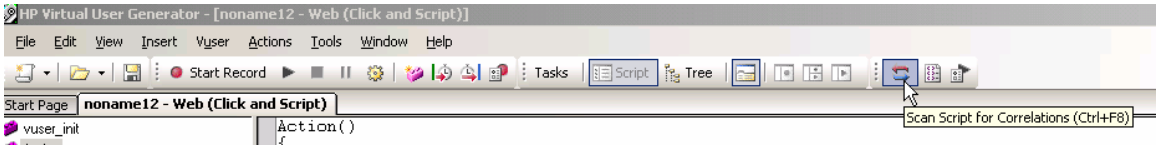
For example, if you perform a series of dependent queries one after another, your code may become very long. To reduce the size of the code, you can nest the queries, but then you lose precision and the code becomes complex and difficult to understand. Correlating the statements enables you to link queries without nesting.

To generate dynamic data

Many applications and Web sites identify a session by the current date and time. If you try to replay a script, it will fail because the current time is different than the recorded time. Correlating the data enables you to save the dynamic data and use it throughout the scenario run.

The main steps in correlating a script are:

1 After Running the script once click on can for correlation button



2 Determine which value to correlate.

For most protocols, you can view the problematic statements in the Execution log. You double-click an error message and jump directly to its location. Alternatively, you can use the WDiff utility distributed with VuGen to determine the inconsistencies within your script. For more information, see

3 Save the results.

You save the value of a query to a variable using the appropriate function. The correlating functions are protocol-specific. Correlation function names usually contain the string save_param, such as web_reg_save_param and lrs_save_param. Refer to the specific protocol

chapters for an explanation on how to perform correlation. In several protocols, such as database and Web, VuGen automatically inserts the functions into your script.

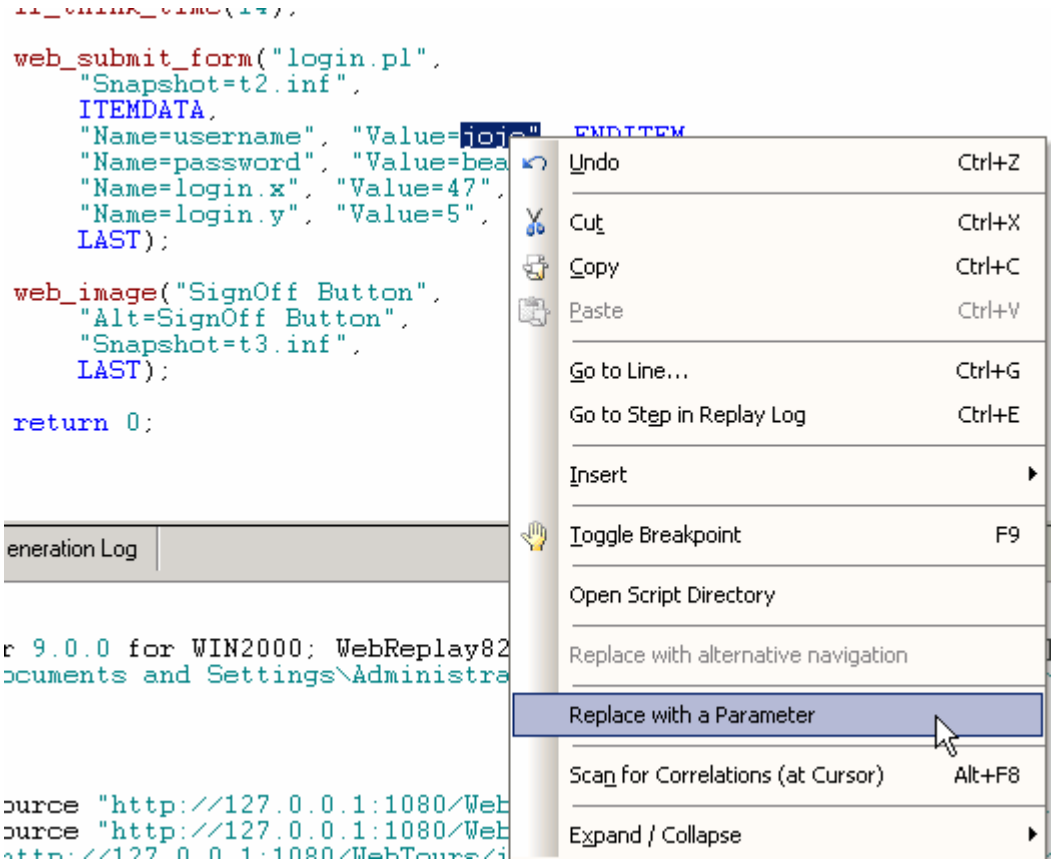
4 Reference the saved values.

Replace the constants in the query or statement with the saved variables.

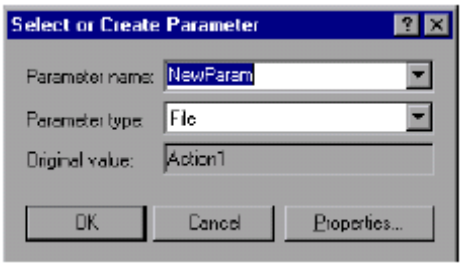
Now you can Insert Parameterization Statements

Steps for parameterization:

- Locate the argument you want to parameterize
- Give the parameter a name
- Select a parameter type
- Define properties for the parameter type
- Replace the argument with a parameter
-



Once you click like the above one you need to give the name for the parameter as in below one

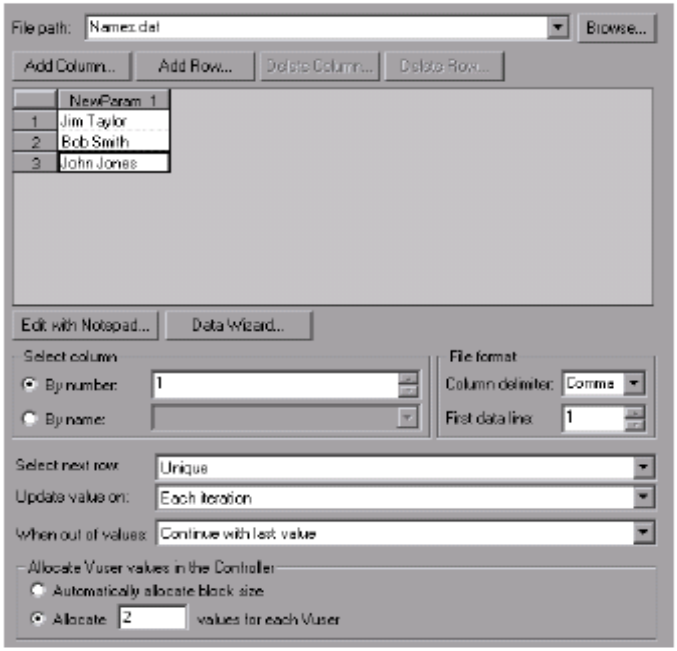


Then Select a parameter type.

Define properties for the parameter type.

Replace the argument with the parameter.

Once you do it the parameterization window will look like this.



Now you can insert rendezvous point in the script

When performing load testing, you need to emulate heavy user load on your system. To accomplish this, you synchronize Vusers to perform a task at exactly the same moment. You configure multiple Vusers to act simultaneously by creating a *rendezvous point*. When a Vuser arrives at the rendezvous point, it waits until all Vusers participating in the rendezvous arrive. When the designated number of Vusers arrive, the Vusers are released.

To insert a rendezvous point:

1 While recording a Vuser script, click the **Rendezvous** button on the Recording toolbar. The Rendezvous dialog box opens.

2 Type a name for the rendezvous point in the **Rendezvous Name** box.

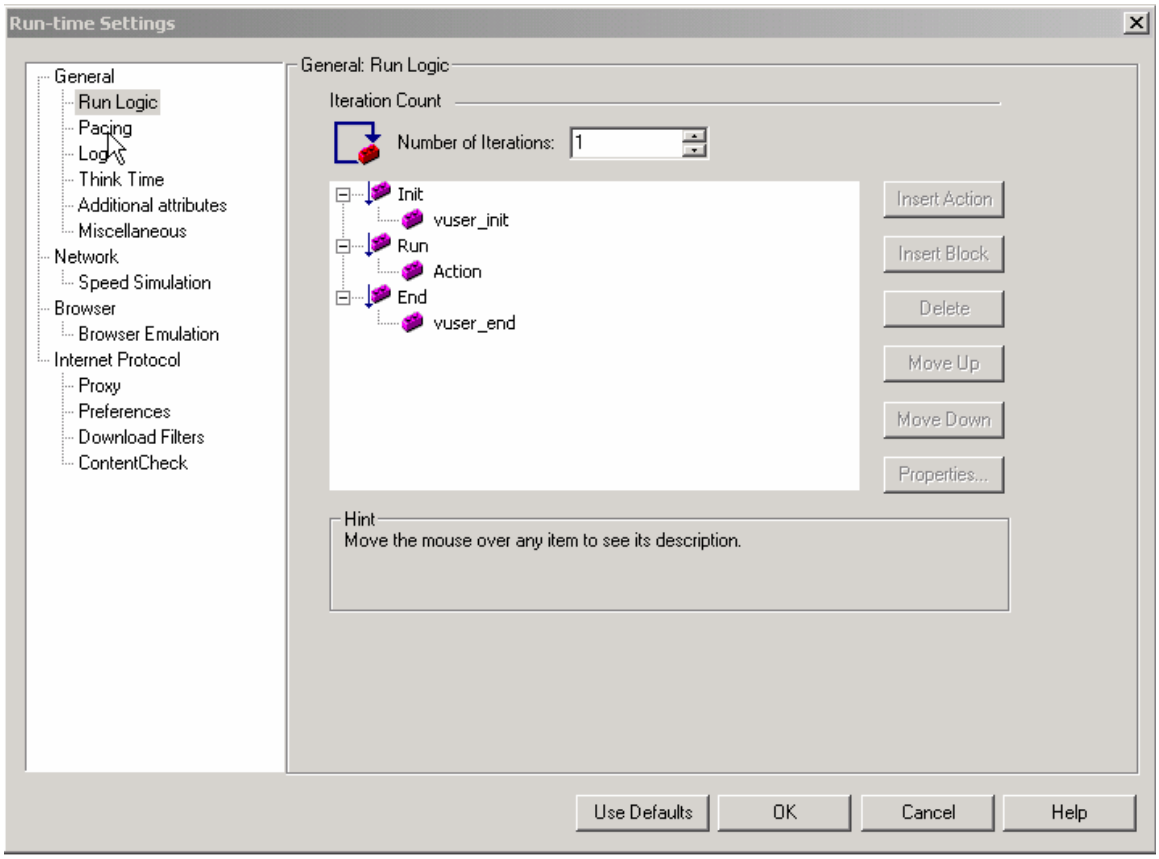
Note: The name for the rendezvous point is not case sensitive. For example, the Vuser recognizes Rendezvous1 and rendezvous1 as the same point.

Click OK. VuGen inserts lr_rendezvous into the Vuser script. For example, the following function defines a rendezvous point named rendezvous1:

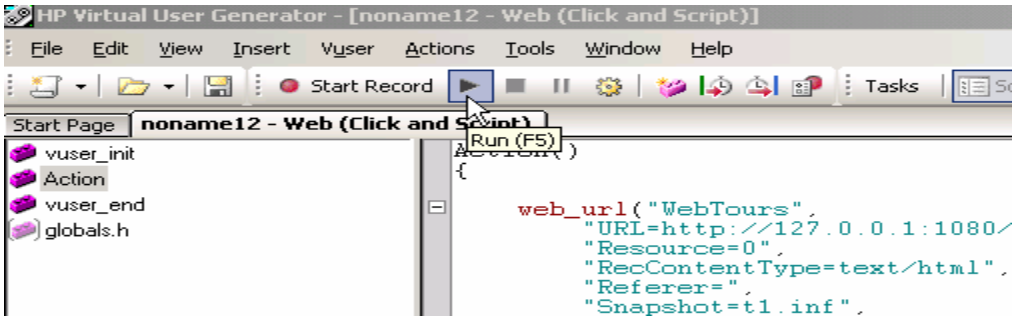
```
lr_rendezvous("rendezvous1");
```

3 To insert rendezvous points into your script after the recording session, select **Insert** > **Rendezvous** from the VuGen toolbar.

Now you can set the run time settings



Now we can run the script in stand alone mode to check if the scripts are running properly



Once the scripts are running fine in stand alone mode , these scripts can be placed in controller for funning the load test.

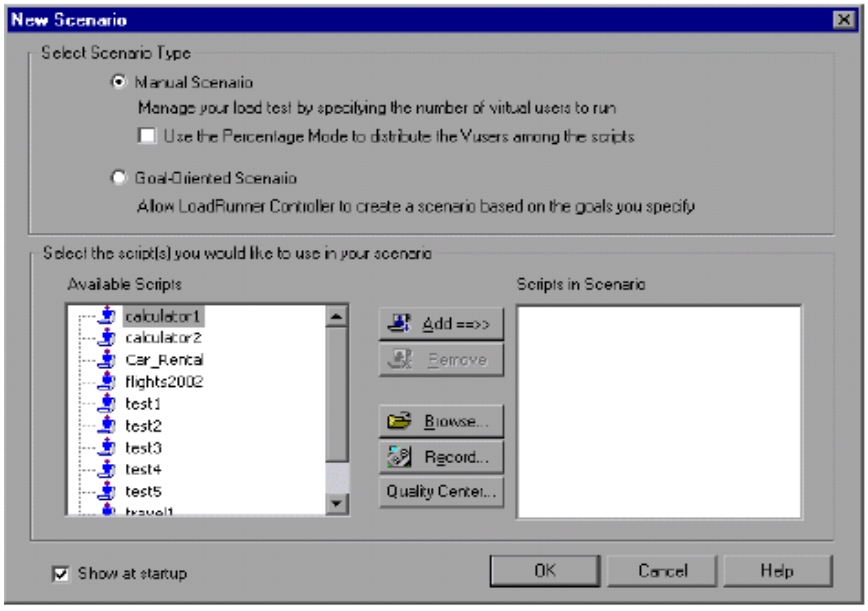
Controller

To open the Controller:

You can open the Controller by choosing either of the following:

- Start > Programs > LoadRunner > Applications > Controller.
- Start > Programs > LoadRunner > LoadRunner. The HP LoadRunner
- launcher window opens. Select the Load Testing tab, and then click Run Load Tests.

If the New Scenario dialog box does not open on startup, you can open it by choosing **File** > **New** or by clicking the **New** button on the Controller toolbar.



Once the Scenario is opened we can do the following

1) Controlling the number of Virtual Users
2) Setting up the SLA
3) Define the group name, Vuser quantity, load generators, and scripts for the Vuser group
4) Add and configure one or more scripts to the Vuser group
5) Enable or disable a Vuser group for the scenario
6) Remove a Vuser group from the scenario
7) Setting up the Run Duration
8) Setting up Load generators
9) Defining the run pattern by using Ramp up and ramp down

Selecting the Scenario Type

Select one of the following scenario options:

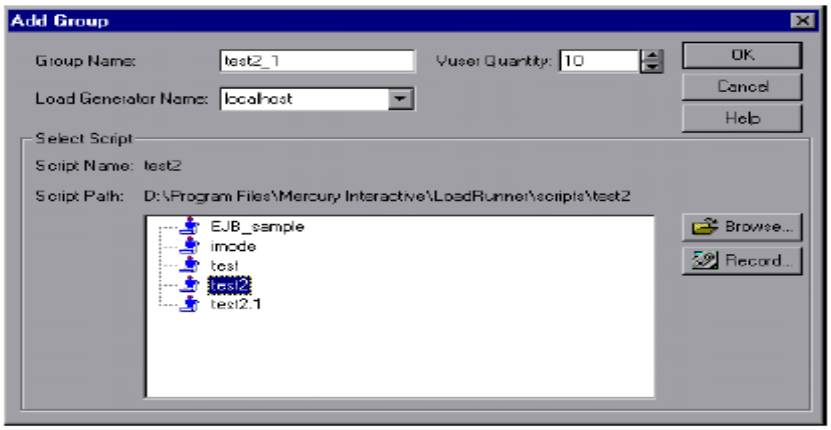
Manual Scenario. Select this method if you want to build a manual scenario. You build a manual scenario by creating groups and specifying the script, the load generator, and the number of Vusers included in each group.

Use the Percentage Mode to distribute the Vusers among the scripts. Select this option if you want to build a manual scenario by specifying a number of Vusers to be distributed among the selected Vuser scripts.

Goal Oriented Scenario. Select this method to have LoadRunner build a scenario for you. In a goal-oriented scenario, you define the goals you want your test to achieve, and LoadRunner automatically builds a scenario for you, based on these goals.

To create Vuser Groups:

1 On the Scenario Groups pane toolbar, click the **Add Group** button. The Add Group dialog box opens:



2 In the Group Name box, enter a name for the Vuser group.

3 From the Vuser Quantity box, select the number of Vusers that you want to create in the group.

Creating Load Generators

Select a load generator from the Load Generator Name list. To use a load generator that does not appear, select **Add** from the Load Generator Name list. The Add New Load Generator dialog box opens:

Type the name of the load generator in the Name box. In the Platform box, select the type of platform on which the load generator is running.

Defining SLA:

An SLA is a formally negotiated agreement between two parties. It is a contract that exists between customers and their service provider, client or between service providers. It records the common understanding about services, priorities, responsibilities, guarantee, and such — collectively, the *level of service*. For example, it may specify the levels of availability, serviceability, performance, operation, or other attributes of the service like billing and even penalties in the case of violation of the SLA.

About Defining Service Level Agreements

When you design a load test scenario, you can define goals or service level agreements (SLAs) for the performance metrics. When you run the scenario, LoadRunner gathers and stores performance-related data. When you analyze the run, Analysis compares this data against the SLAs and determines SLA statuses for the defined measurements.

Depending on the measurements that you are evaluating, LoadRunner
Determines SLA statuses in one of the following ways:

SLA status determined per time interval within the run. Analysis displays SLA statuses at set time intervals—for example, every 10 seconds—over a timeline within the run.

SLA status determined over the whole run. Analysis displays a single SLA status for the whole scenario run.

To define an SLA for Average Transaction Response Time:

1 Open the Service Level Agreement Wizard.

- ➤ **If you are working in Analysis:** Select **Tools** > **Configure SLA Rules** to open the Service Level Agreement dialog box. Click **New**.
- ➤ **If you are working in the Controller:** In the Design tab, in the Service Level Agreement pane, click **New**.

The Service Level Agreement wizard opens.

2 Complete the SLA Wizard steps, as described in the following sections.

Measurement—Select a Measurement for Your Goal

For Average Transaction Response Time, LoadRunner evaluates SLA statuses at set time intervals within the run.

Under **SLA status determined per time interval within the run**, select **Average Transaction Response Time**.

Transactions—Select Transactions

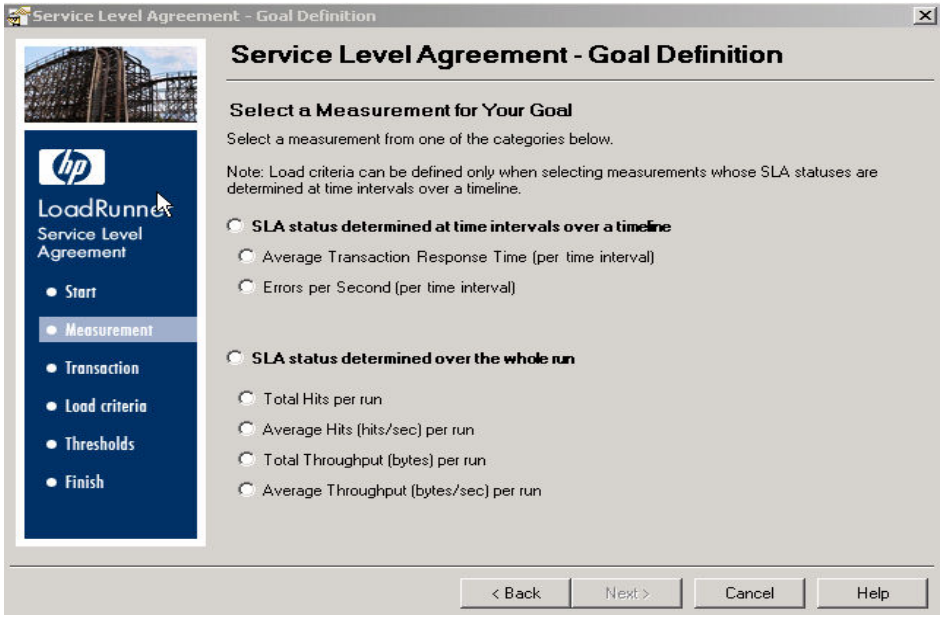
From the **Available Transactions** list, select the transactions that you want to evaluate as part of your SLA and click **Add**.

Note: You can select multiple transactions using the CTRL key.

The transactions you selected are displayed in the **Selected Transactions** list.

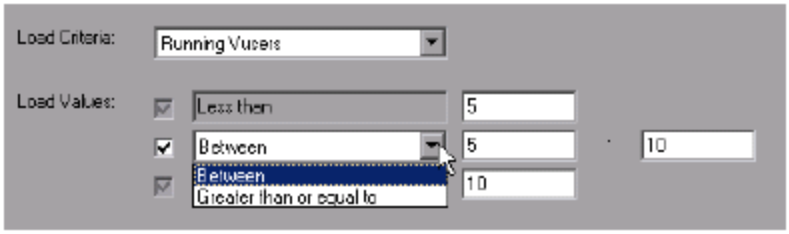
Load Criteria—Set Load Criteria

Select load criteria for your goal and define appropriate load value ranges.


Service Level Agreement - Goal Definition
LoadRunner
Service Level Agreement
Start
Measurement
Transaction
Load criteria
Thresholds
Finish
Service Level Agreement - Goal Definition
Select a Measurement for Your Goal
Select a measurement from one of the categories below.
Note: Load criteria can be defined only when selecting measurements whose SLA statuses are determined at time intervals over a timeline.
SLA status determined at time intervals over a timeline
Average Transaction Response Time (per time interval)
Errors per Second (per time interval)
SLA status determined over the whole run
Total Hits per run
Average Hits (hits/sec) per run
Total Throughput (bytes) per run
Average Throughput (bytes/sec) per run
< Back
Next >
Cancel
Help

- In the **Load Criteria** box, select the relevant load criteria that you want to use—for example, **Running Vusers**.

  To define an SLA without load criteria, select **None** in the **Load Criteria** box.
- In the **Less than** box, set the lower load value range by entering a maximum value for this range. This range is between 0 and the maximum value you entered, but does not include the maximum value.
- To set in-between load value ranges, select **Between** from the **Greater than or equal to/Between** combo box and enter minimum and maximum values for the range. The minimum value is included in the range and the maximum value is not.



Note: You can set up to three in-between ranges.

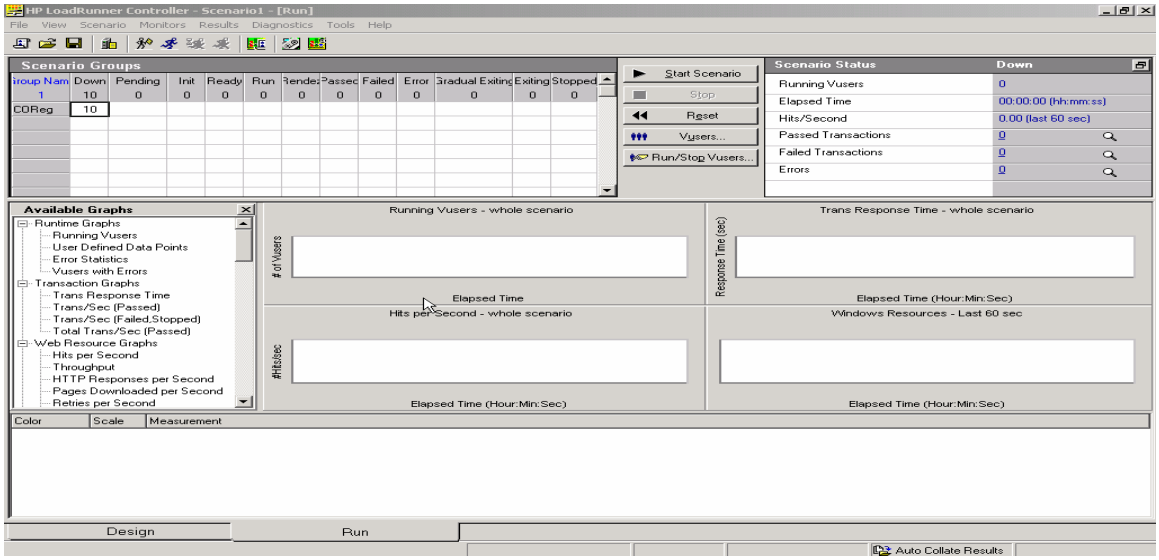
- To set the upper load value range, select **Greater than or equal to** from the **Greater than or equal to/Between** combo box and enter the minimum value for this range. The minimum value is included in this range.

Note: Valid load value ranges are consecutive—there are no gaps in the range—and span all values from zero to infinity.

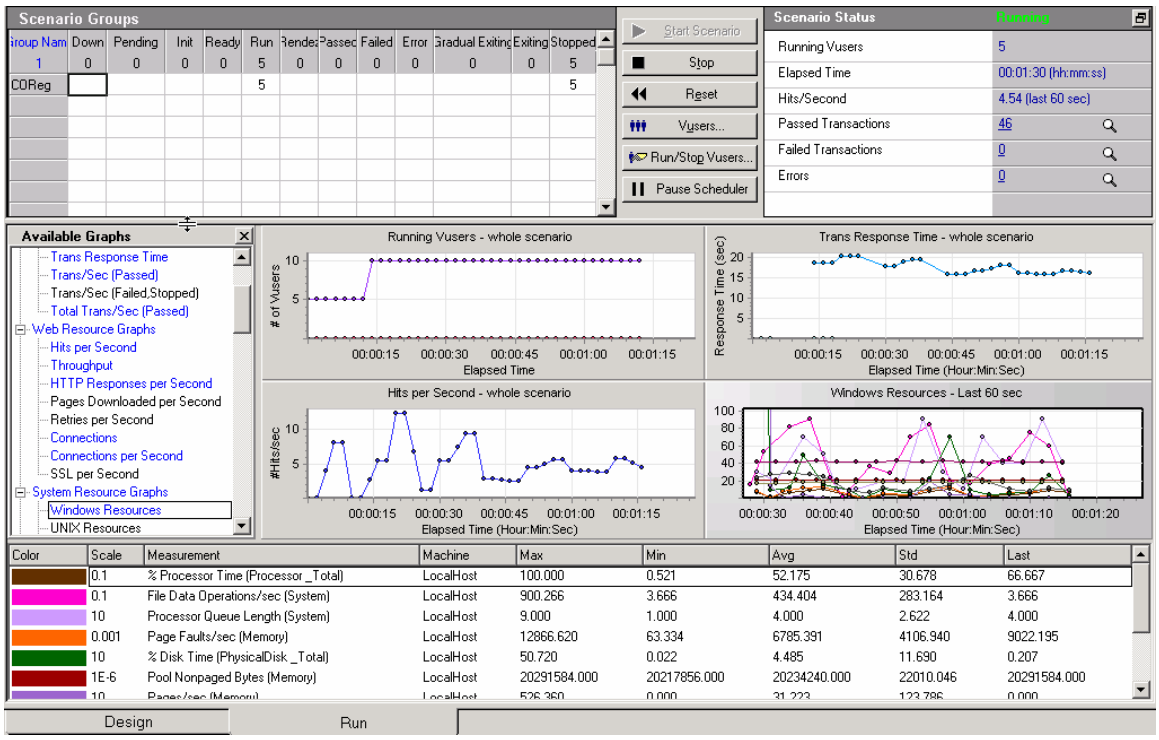
Running the Scenario

The following procedure outlines how to run a scenario:

- Open an existing scenario or create a new one.
- Configure and schedule the scenario.
- Set the results directory.
- Run and monitor the scenario.



And once you run the scenario you will get the details like this



Graph Analysis:

During load test scenario execution, Vusers generate result data as they perform their transactions. The Analysis graphs help you determine system performance and provide information about transactions and Vusers. You can compare multiple graphs by combining results from several load test scenarios or merging several graphs into one.

The Report utilities enable you to view a Summary HTML report for each graph or a variety of Performance and Activity reports. You can create a report as a Microsoft Word document, which automatically summarizes and displays the test's significant data in graphical and tabular format.

Analysis graphs are divided into the following categories:

- Vuser Graphs. Provide information about Vuser states and other Vuser statistics.
- Error Graphs. Provide information about the errors that occurred during the load test scenario.
- Transaction Graphs. Provide information about transaction performance and response time.
- Web Resource Graphs. Provide information about the throughput, hits per second, HTTP responses per second, number of retries per second, and downloaded pages per second for Web Vusers.
- Web Page Diagnostics Graphs. Provide information about the size and download time of each Web page component.
- System Resource Graphs. Provide statistics relating to the system resources that were monitored during the load test scenario using the online monitor.

Starting Analysis

You can open Analysis as an independent application or directly from the Controller. To open Analysis as an independent application, choose one of the following:

- Start > Programs > LoadRunner > Applications > Analysis
- Start > Programs > LoadRunner > LoadRunner, select the Load Testing tab, and then click Analyze Load Tests.

Once you start the Session a Summary report will be displayed

Summary Report:

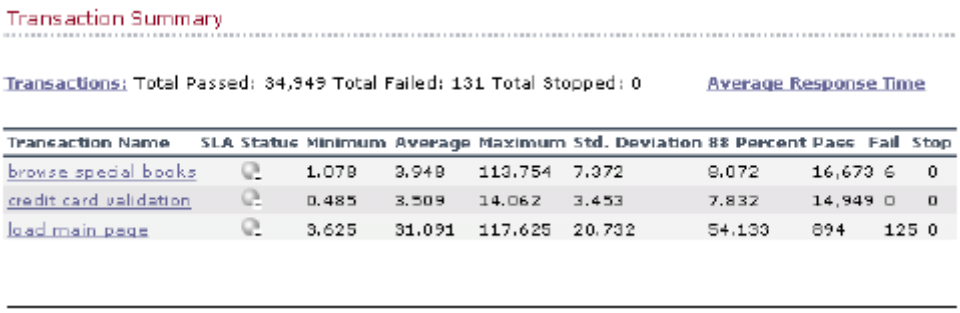
The Summary report provides general information about load test scenario execution. This report is always available from the Session Explorer or as a tab in the Analysis window. The Summary report lists statistics about the scenario run and provides links

to the following graphs: Running Vusers, Throughput, Hits Per Second, HTTP Responses per Second, Transaction Summary, and Average Transaction Response Time. The appearance of the Summary report and the information displayed will vary depending on whether an SLA (Service Level Agreement) was defined. An SLA defines goals for the scenario. LoadRunner measures these goals during the scenario run and analyzes them in the Summary report.

This Summary Report will have the following details:

Transaction Summary

This section displays a table containing the load test scenario's diagnostics data. Included in this data is a percentile column (x Percent). This column indicates the maximum response time for that percentage of transactions performed during the run. For example, in the table below, the value in the 88 Percent column for browse special books is 8.072. This means that the response time for 88% of the browse special books transactions was less those 8.072 seconds.



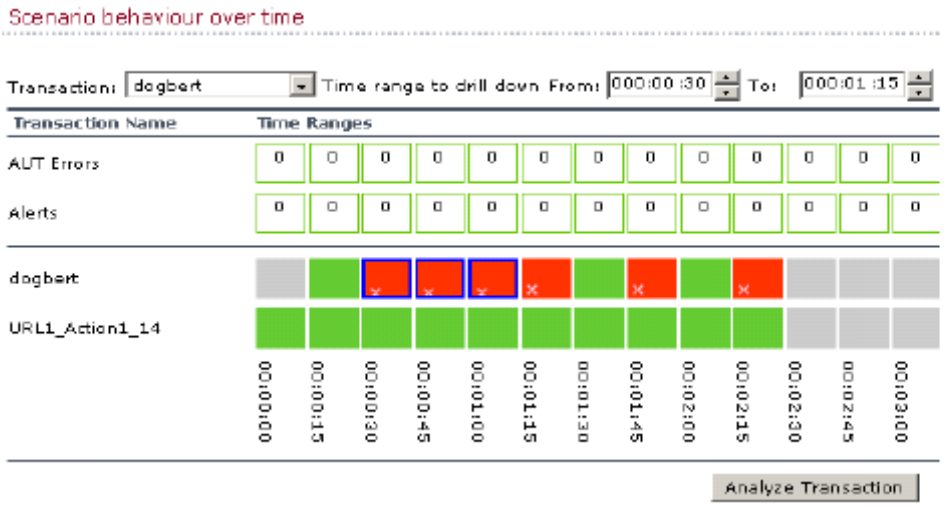
Transaction Summary

Transactions: Total Passed: 34,949 Total Failed: 131 Total Stopped: 0 Average Response Time

Transaction Name	SLA Status	Minimum	Average	Maximum	Std. Deviation	88 Percent	Pass	Fail	Stop
browse special books		1.078	3.949	113.754	7.372	8.072	16,673	6	0
credit card validation		0.485	3.509	14.062	3.453	7.832	14,949	0	0
load main page		3.625	31.091	117.625	20.732	54.133	894	125	0

Scenario Behavior over Time

This section displays the average errors per second received by the application under test per time interval. For example, 0 means that on average there were zero errors received per second for that time interval, 0+ means that on average there were slightly more than zero errors received, and so on.

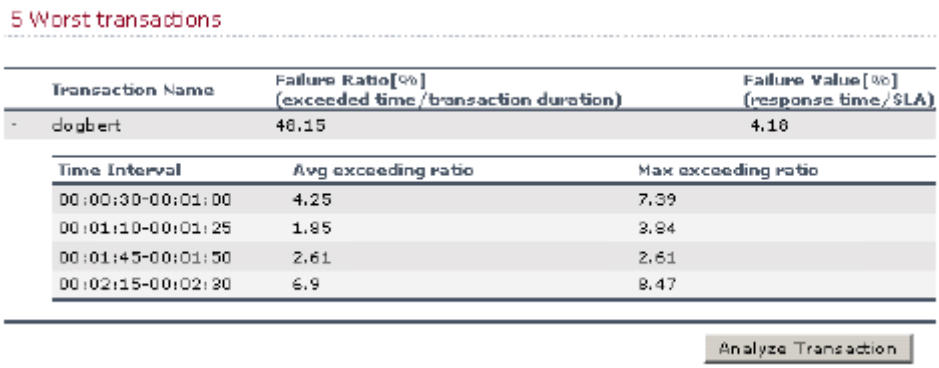


HTTP Responses Summary

This section shows the number of HTTP status codes returned from the Web server during the load test scenario, grouped by status code.

X Worst Transaction

The X Worst Transactions table shows the worst transactions in terms of how often the transactions exceeded the SLA boundary during the run, and by how much.



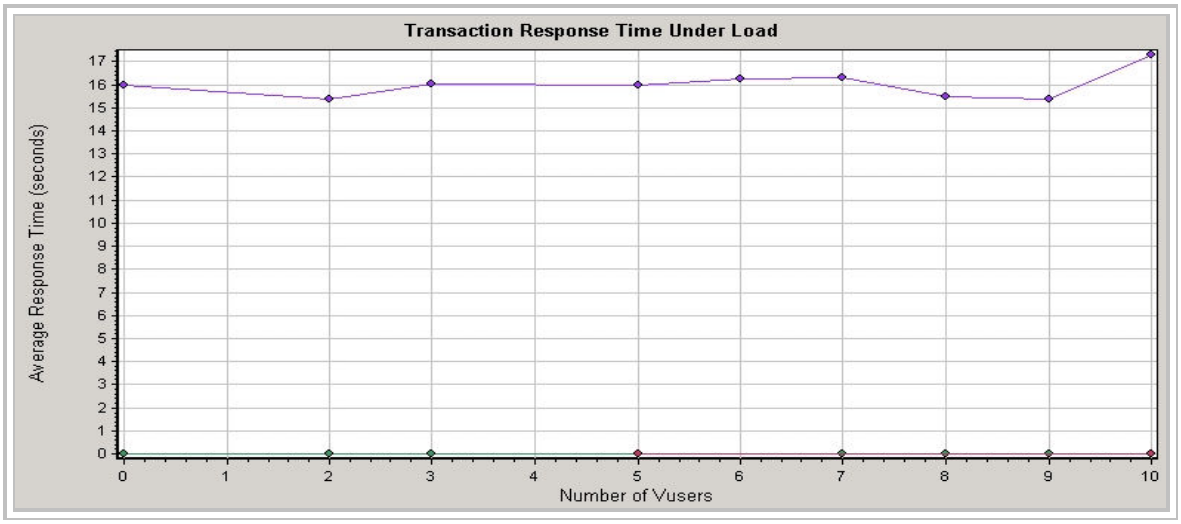
SLA Report

An SLA (Service Level Agreement) defines goals for the load test scenario. LoadRunner measures these goals during the scenario run and analyzes them in the Summary report. The SLA Report shows the succeeded or failed status of all SLAs that were defined for the scenario run.

The Following are the other Graphs that are Displayed

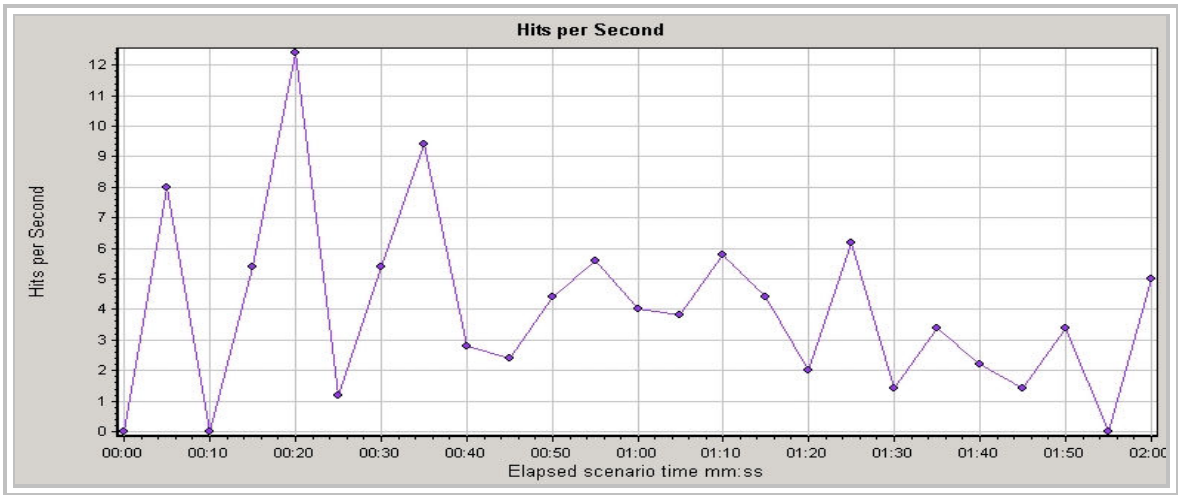
Transaction Response Time under Load

Displays average transaction response times relative to the number of Vusers running at any given point during the load test. This graph helps you view the general impact of Vuser load on performance time and is most useful when analyzing a load test which is run with a gradual load.



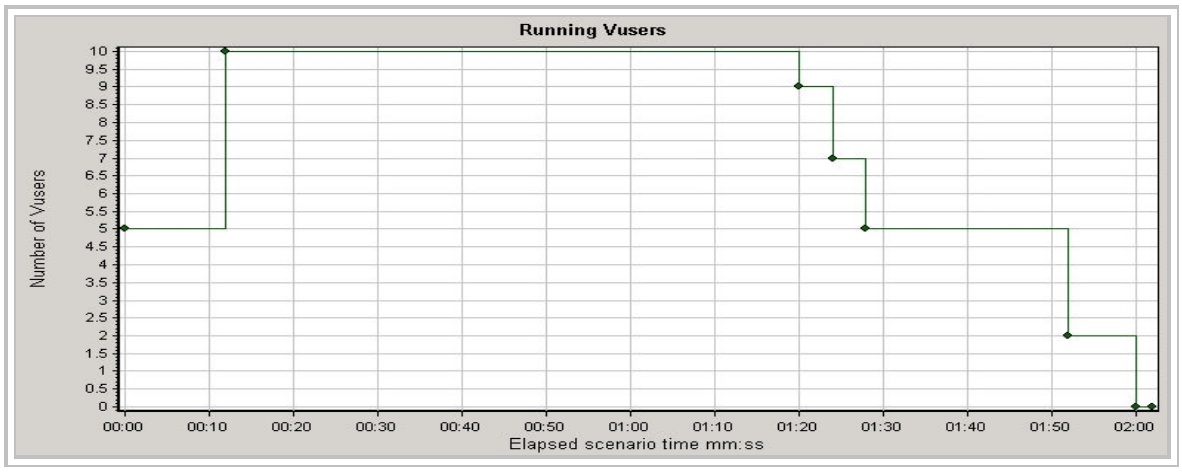
Hits Per Second Graph

Displays the number of hits made on the Web server by Vusers during each second of the load test. This graph helps you evaluate the amount of load Vusers generate, in terms of the number of hits.



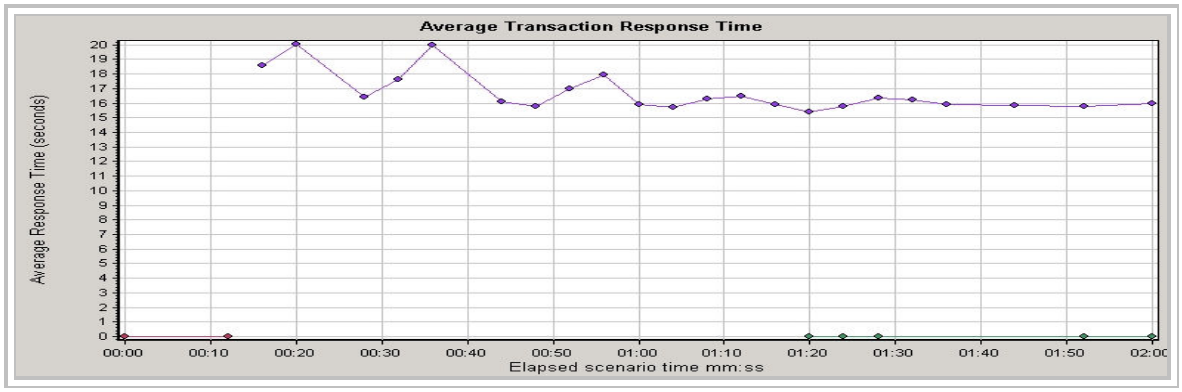
Running Vusers Graph

Displays the number of Vusers that executed Vuser scripts, and their status, during each second of a load test. This graph is useful for determining the Vuser load on your server at any given moment.



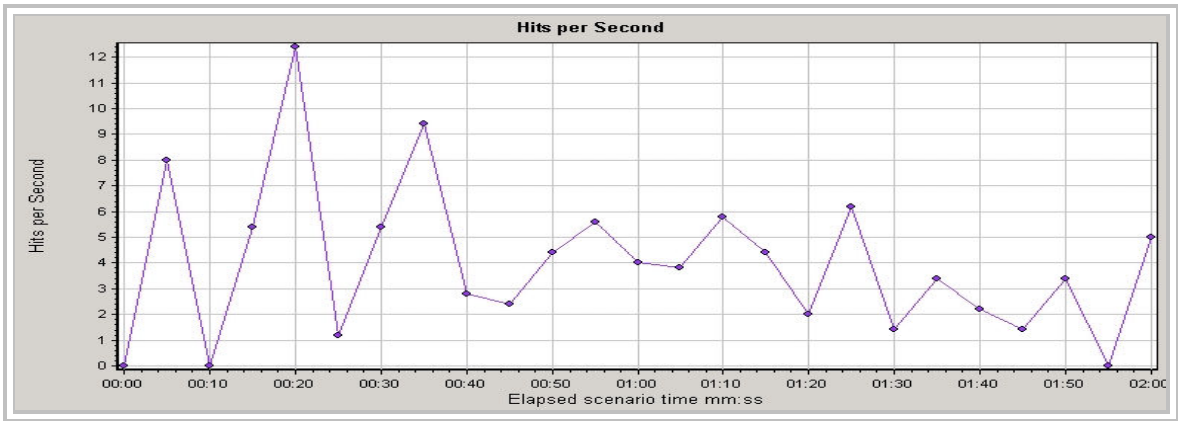
Average Transaction Response Time

Displays the average time taken to perform transactions during each second of the load test. This graph helps you determine whether the performance of the server is within acceptable minimum and maximum transaction performance time ranges defined for your system.



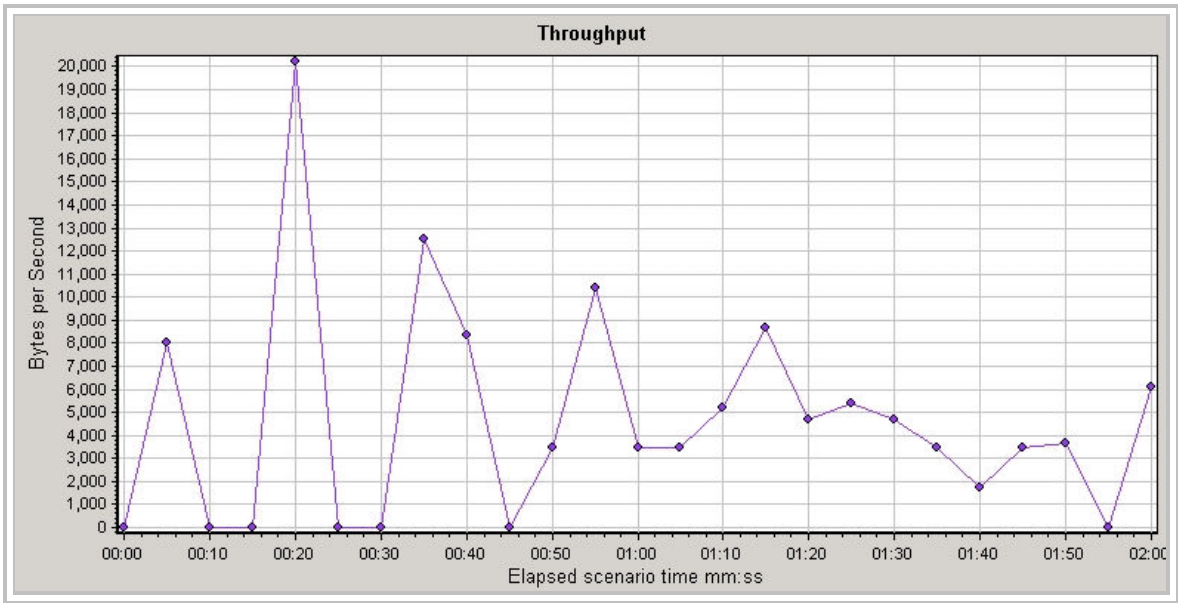
Hits per Second

Displays the number of hits made on the Web server by Vusers during each second of the load test. This graph helps you evaluate the amount of load Vusers generate, in terms of the number of hits.



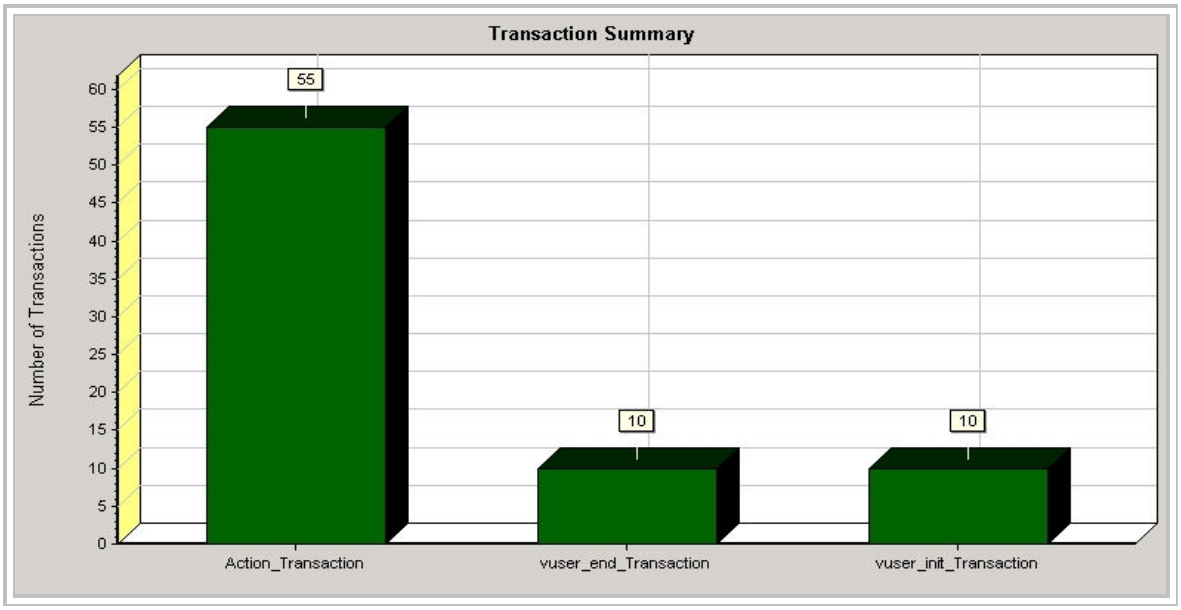
Throughput

Displays the amount of throughput (in bytes) on the Web server during the load test. Throughput represents the amount of data that the Vusers received from the server at any given second. This graph helps you to evaluate the amount of load Vusers generate, in terms of server throughput.



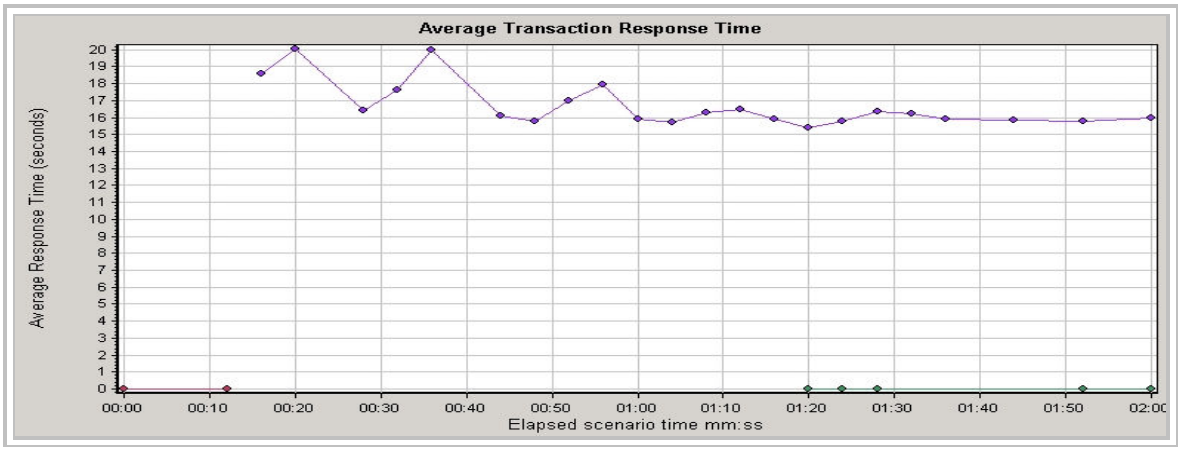
Transaction Summary

Displays the number of transactions that passed, failed, stopped, or ended with errors.



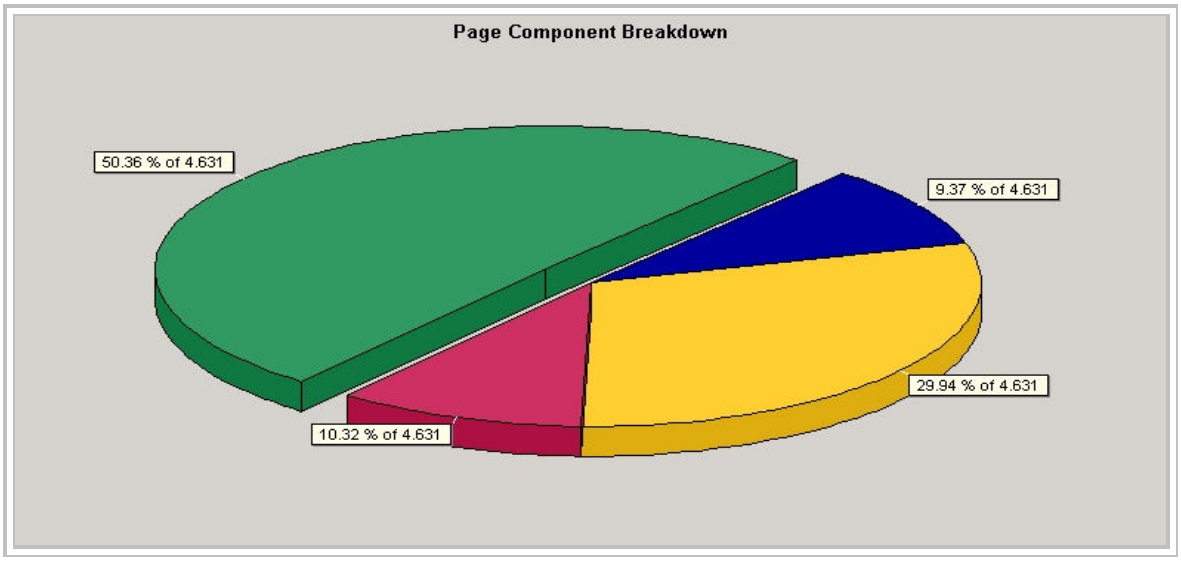
Average Transaction Response Time

Displays the average time taken to perform transactions during each second of the load test. This graph helps you determine whether the performance of the server is within acceptable minimum and maximum transaction performance time ranges defined for your system.



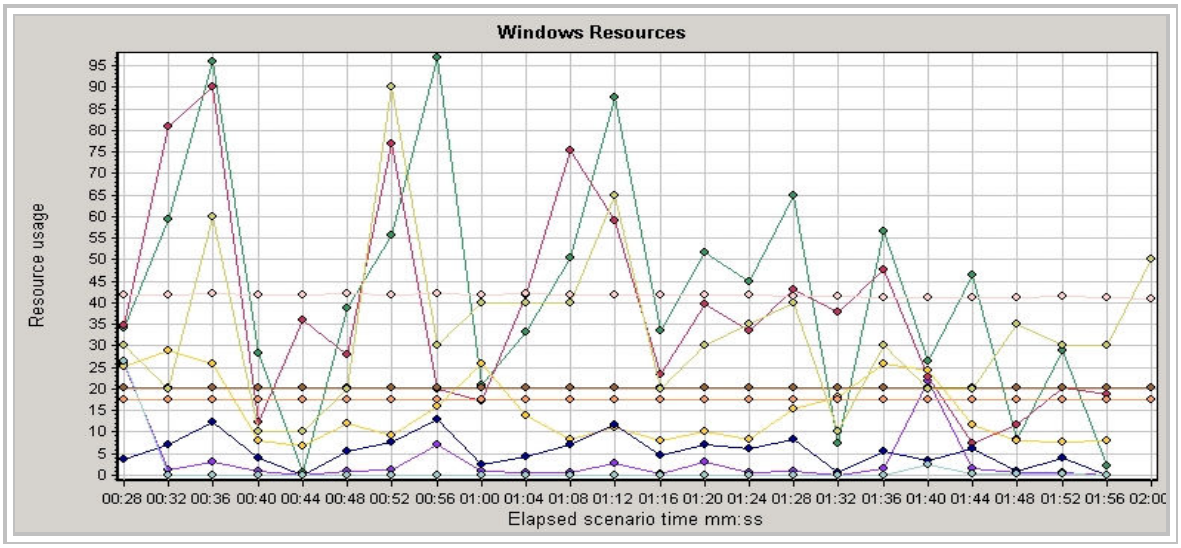
Page Component Breakdown

Displays the average download time (in seconds) for each Web page and its components.



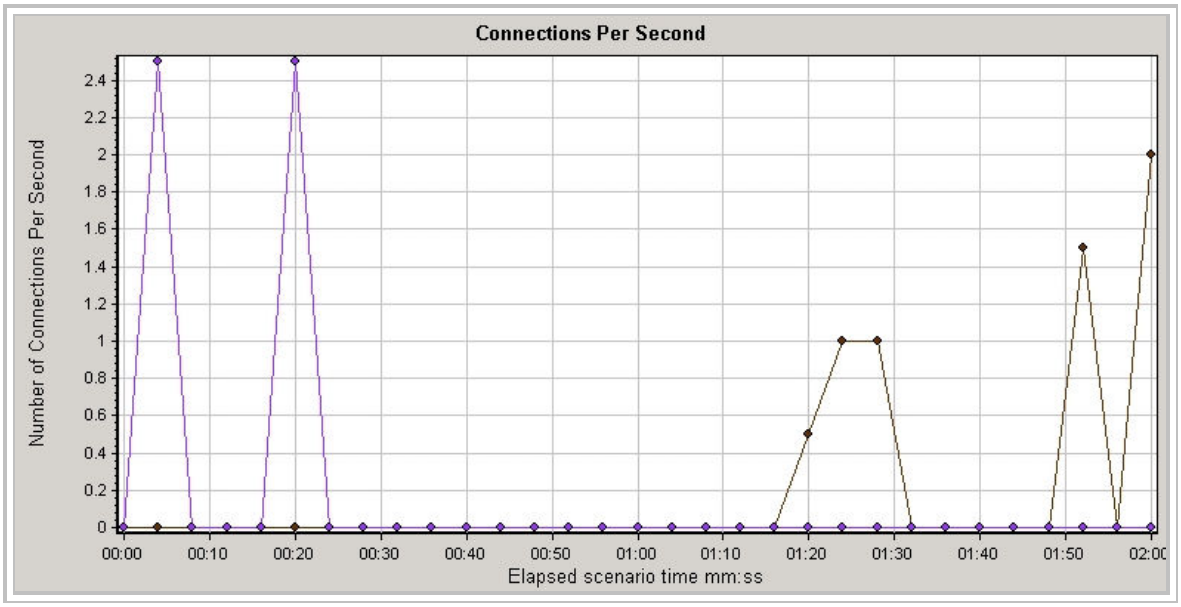
Windows Resources

Displays a summary of the System Resources usage for each Windows based host.



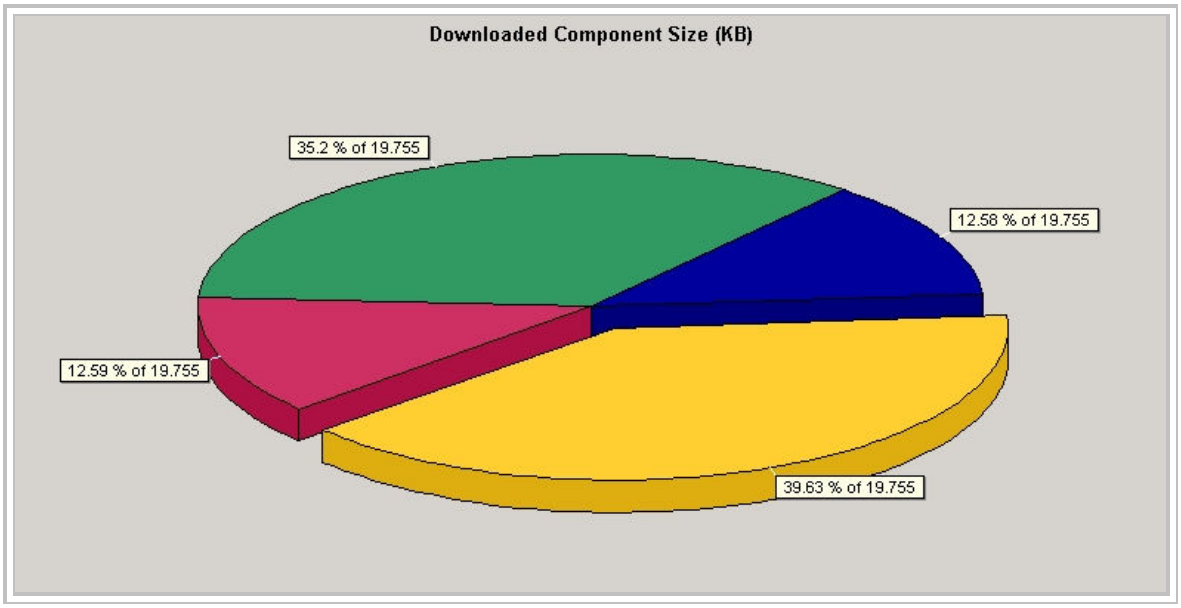
Connections Per Second

Displays the number of Connections per Second



Downloaded Component Size (KB)

Displays the size of each Web page component. Note that the size displayed includes both the size of the component and its header.



Time to First Buffer Breakdown

Displays each Web page component's relative server/network time (in seconds) for the period of time until the first buffer is successfully received back from the Web server.