

Rational Performance Test Server



# Getting Started Guide

*Version 8.0.0*



**Note**

Before using this information and the product it supports, read the information in “Notices” on page 44.

This edition applies to version 8.0.0 of Rational Performance Test Server and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2001, 2012.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>About this Publication</b> .....	<b>v</b>
Intended Audience .....	vi
Scope .....	vi
Typographical Conventions .....	vi
Contacting IBM Support .....	vi
<b>Overview</b> .....	<b>1</b>
Rational Performance Test Server Components .....	2
Prerequisites .....	3
Quick Start Summary .....	4
<b>Setting up a Project</b> .....	<b>5</b>
Creating a Project .....	6
Creating an Agent .....	7
<b>Creating a Simple Test</b> .....	<b>9</b>
Overview .....	10
Creating a Test .....	12
<b>Creating Performance Tests</b> .....	<b>13</b>
Creating a Performance Test .....	14
Adding a Distributed Test .....	16
Selecting a Test Engine .....	18
<b>Running Performance Tests</b> .....	<b>19</b>
Running the Agent .....	20
Executing the Test .....	21

---

<b>Viewing Results</b> .....	<b>22</b>
The Results Gallery .....	23
Plotting Multiple Tests .....	27
Saving and Loading Charts .....	29
<b>Adding Probes</b> .....	<b>30</b>
Adding a Probe .....	31
Configuring the Agent .....	32
Modifying Statistics Settings .....	33
<b>Additional Options</b> .....	<b>34</b>
Mapping Tag Values with Data Sets .....	35
Chart Options .....	36
Configuring Test Loads .....	38
<b>Glossary</b> .....	<b>42</b>
<b>Notices</b> .....	<b>44</b>
Trademarks and service marks .....	47

# About this Publication

## **Contents**

### **Intended Audience**

### **Scope**

### **Typographical Conventions**

### **Contacting IBM Support**

This document is a tutorial to introduce the reader to IBM® Rational® Performance Test Server. It takes you through some worked examples to make you familiar with creating performance tests and analyzing the results.

For an introduction to the concepts and components involved in performance testing, refer to *IBM Rational Performance Test Server Reference Guide*.

---

## Intended Audience

Readers should already be familiar with using IBM Rational Integration Tester for functional testing, in particular the concepts of tests, environments, test data sets, and general product usage are assumed. These same tests are used within Rational Performance Test Server. Further details on these concepts can be found in *IBM Rational Integration Tester Getting Started Guide* and the *IBM Rational Integration Tester Reference Guide*.

## Scope

This document is concerned only with introducing readers to IBM Rational Performance Test Server, in particular running performance tests. It should be read in conjunction with *IBM Rational Performance Test Server Reference Guide*, which provides full details about using the software.

For information about installing Rational Performance Test Server, refer to *IBM Rational Performance Test Server Installation Guide*.

## Typographical Conventions

The following typographical conventions are observed throughout this document:

---

Type	Usage
Constant Width	Program output, listings of code examples, file names, commands, options, configuration file parameters, and literal programming elements in running text.
<i>Italic</i>	Document title names in statements that refer you to other documents. Also used to highlight concepts when first introduced.
<b>Bold</b>	Menu items in graphical user interface windows (such as Microsoft Windows-based or UNIX X Window applications) from which you select options or execute macros and functions.  Submenus and options of a menu item are indicated with a “greater than” sign, such as <b>Menu &gt; Submenu</b> or <b>Menu &gt; Option</b> .

---

## Contacting IBM Support

To contact IBM Support, see: [www.ibm.com/contact/us/en/](http://www.ibm.com/contact/us/en/)

---

# Overview

## **Contents**

**Rational Performance Test Server  
Components**

**Prerequisites**

**Quick Start Summary**

This chapter provides an overview of Rational Performance Test Server and the contents of this guide.

---

## 1.1 Rational Performance Test Server Components

The main components of Rational Performance Test Server are as follows:

- Rational Performance Test Server GUI - used to create, edit and analyzes results of performance tests.
- Performance Test Controller - Rational Performance Test Server instance used to initiate performance tests at run time, may be via the GUI or on the command line
- Agent - process that launches Test Engines and Probes
- Test Engine - instance of Rational Integration Tester executing performance tests
- Probe - program monitoring infrastructure components such as servers and processes.

---

## 1.2 Prerequisites

The following sections list the tasks and required system components that should be completed or configured before continuing with this guide.

- [Product Installation](#)
- [Project Database](#)

### 1.2.1 Product Installation

Rational Performance Test Server must be installed on each machine that is to be used for performance testing. This includes systems that execute the tests, and those that monitor the servers using probes.

For information about setup requirements and installation tasks, refer to *IBM Rational Performance Test Server Installation Guide*.

### 1.2.2 Project Database

All performance test data is stored in the project database, so you will need to have access to one. Details about setting up the database and enabling access to it can be found in *IBM Rational Integration Tester Installation Guide*.

If necessary, seek the assistance of your local database administrator. Further scripts may be required, depending on which probes are to be used

---

## 1.3 Quick Start Summary

Throughout this guide, the following steps will be carried out as you create and execute your first performance tests:

1. Create a simple test that includes timed sections
2. Create and configure a performance test
3. Start the Agent process
4. Run the performance test
5. Analyse the results
6. Add optional probes

# Setting up a Project

## **Contents**

**Creating a Project**

**Creating an Agent**

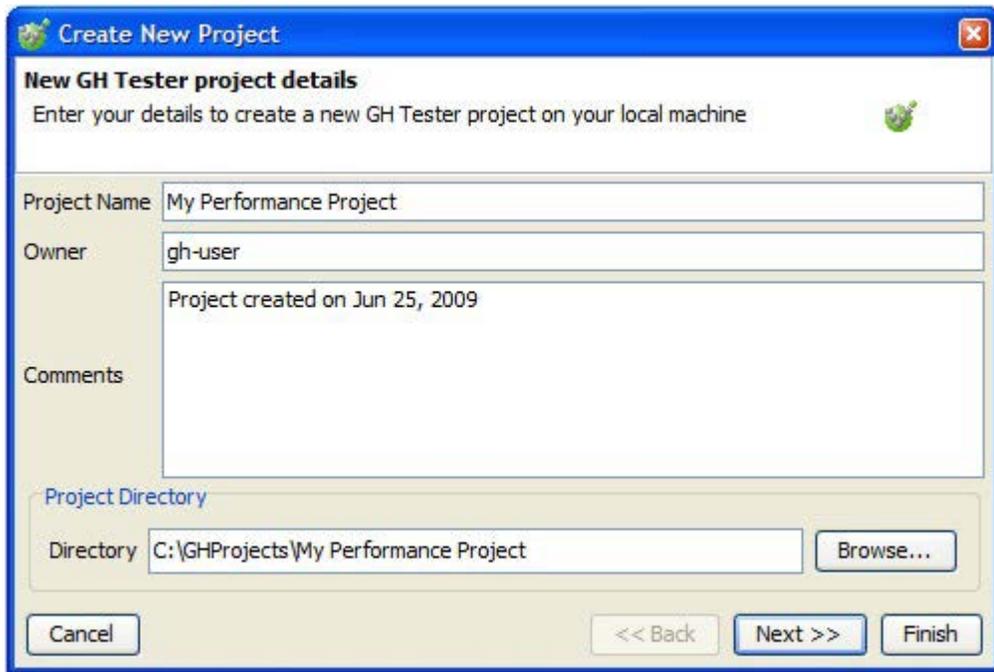
This chapter provides information about how to set up a new project that will be used to create and execute your performance tests.

---

## 2.1 Creating a Project

Before you can start to create and configure a performance test, you must first ensure that your project is configured properly, including the connection to the project database.

1. Launch Rational Performance Test Server and select the **New Project** option in the welcome dialog.
2. Give the new project a name and click **Next** to continue.



The screenshot shows a Windows-style dialog box titled "Create New Project". The main heading is "New GH Tester project details" with a sub-instruction: "Enter your details to create a new GH Tester project on your local machine". The form contains several input fields: "Project Name" with the text "My Performance Project", "Owner" with "gh-user", and "Comments" with "Project created on Jun 25, 2009". Below these is a "Project Directory" section with a text box containing "C:\GHProjects\My Performance Project" and a "Browse..." button. At the bottom of the dialog are four buttons: "Cancel", "<< Back", "Next >>", and "Finish".

3. Provide the details required to connect to a valid project database, and click **Test Connection** to ensure that the configuration and database schema are correct.
4. Once the connection test is successful, click **Finish** to create the new project.
5. Create a new environment, which will be used by default in the new project.

**NOTE:** More information about creating a project and configuring the connection to the project database can be found in *IBM Rational Integration Tester Installation Guide*, *IBM Rational Integration Tester Getting Started Guide*, and *IBM Rational Integration Tester Reference Guide*.

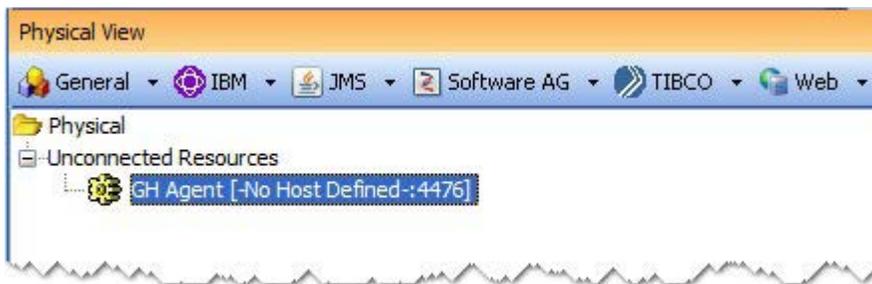
---

## 2.2 Creating an Agent

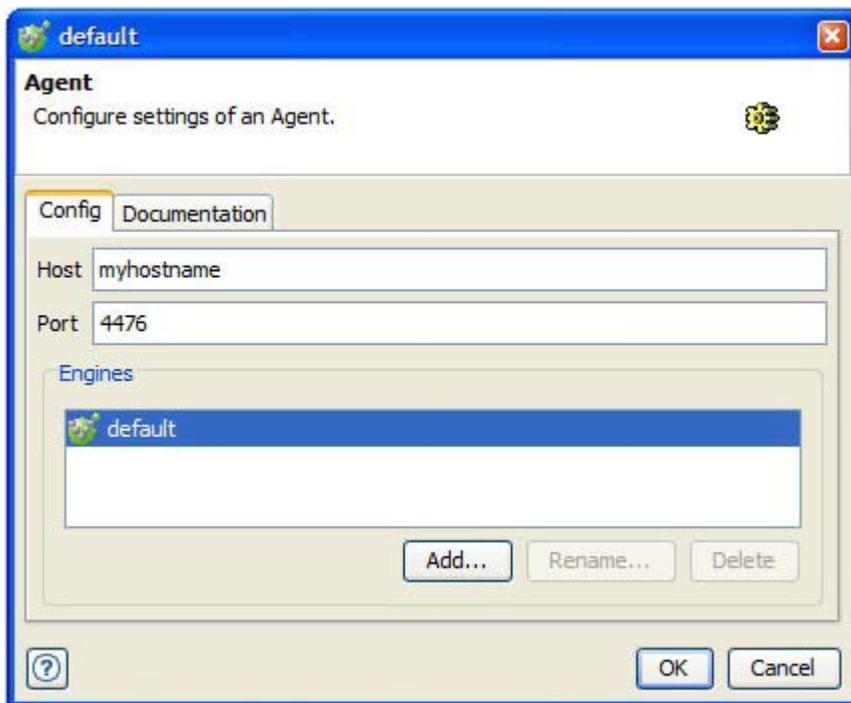
Agents run the actual performance tests, and they are managed in the Physical View of the Architecture School perspective.

1. In the newly opened project, select the Physical View in the Architecture School perspective.
2. Select **General > Agent** from the main toolbar, or right-click the **Physical** folder and select **New > General > Agent** from the context menu.

The new agent will be created under the **Unconnected Resources** node, since no host for the agent has yet been defined.



3. Double-click the new agent to edit it.



- 
4. Enter the host name or IP address of your workstation (that is, the machine where the agent is installed) under **Host**, and leave the port on which the agent is listening unchanged.

**NOTE:** Host names are preferred since an IP address is more likely to change. Also, you should avoid the use of “localhost,” as issues could arise if the agent’s test is run from a different machine.

Every agent includes a default test engine (named **default**) that can not be renamed or deleted.

5. To enter additional notes or information about the agent, select the **Documentation** tab.
6. When you are finished configuring the agent, click **OK**.

The agent is displayed under an appropriate subnet and host in the Physical View.

# Creating a Simple Test

## **Contents**

### **Overview**

### **Creating a Test**

This chapter provides information about how to create a simple test that will be used in a performance test.

---

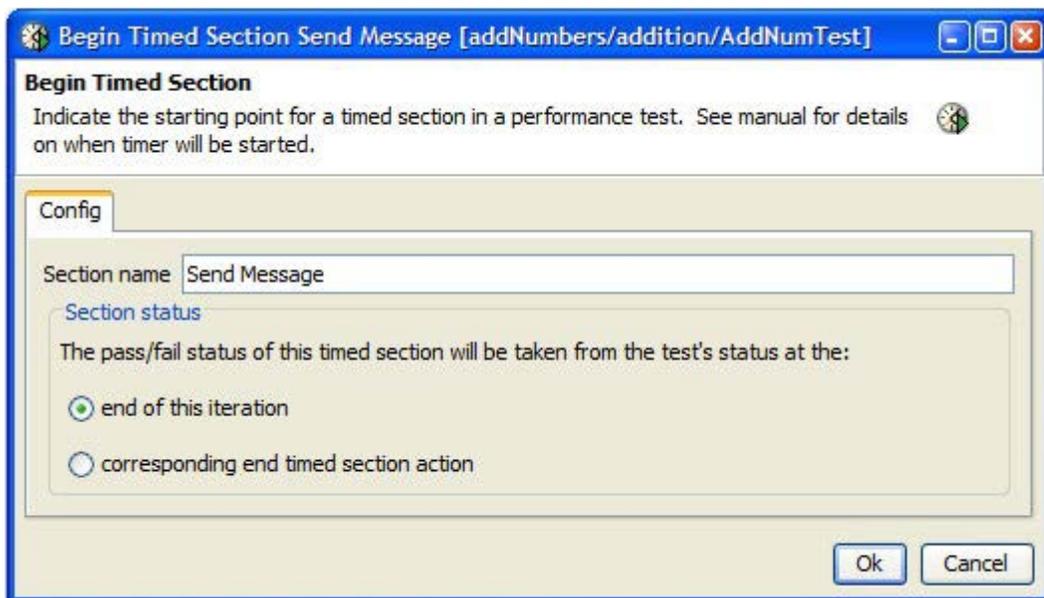
## 3.1 Overview

A timed section is used to analyze a set of steps that are run within a performance test. The timed section is created by adding specific test actions at the start and end of the steps being analyzed.

The following sections describe the test actions that are used to define a timed section. Next, an example is provided that walks you through the steps required to create a test that includes a timed section.

### 3.1.1 Begin Timed Section

The **Begin Timed Section** action marks the start of a timed section within the test.



You can assign a name to the test action to help identify it or to identify the series of steps that it spans. The name of the action will appear in the charting configuration tree, so you should choose a name that is meaningful.

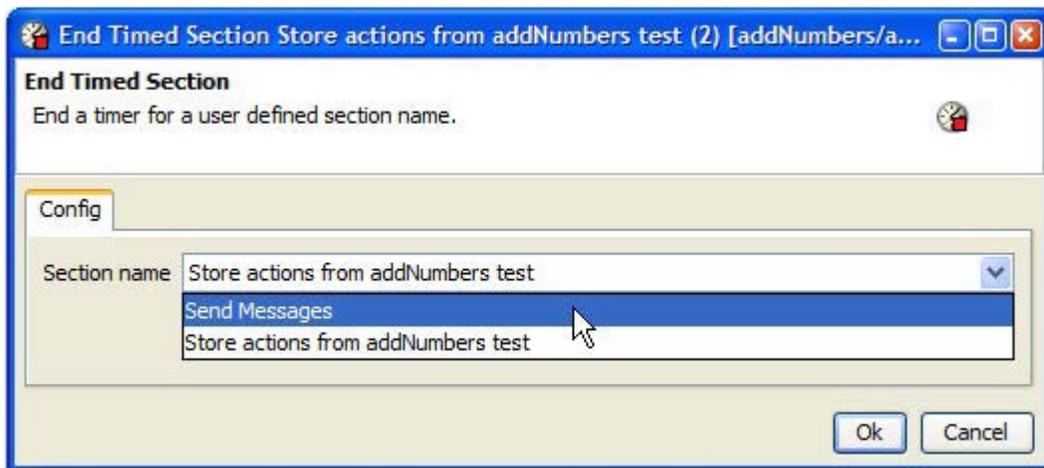
The action will store a pass or fail status based on the status of the test that contains it, which allows you to plot charts based on the outcome of the test iterations. You can store the status of the overall test iteration (**end of this iteration**) or the current status when the timed section ends (**corresponding end timed section action**).

**NOTE:** The status can be different for each of these selections (for example, the test could be passing when the timed section ends, but later actions in the test could cause it to fail).

---

### 3.1.2 End Timed Section

The **End Timed Section** action marks the end of an individually timed section within the test.



There is nothing to configure within the test action, except to select the appropriate action that marks the start of the desired timed section in the test.

**NOTE:** If the test contains more than one timed section, it is important that each **Begin Timed Section** action is terminated with an appropriate **End Timed Section** action.

---

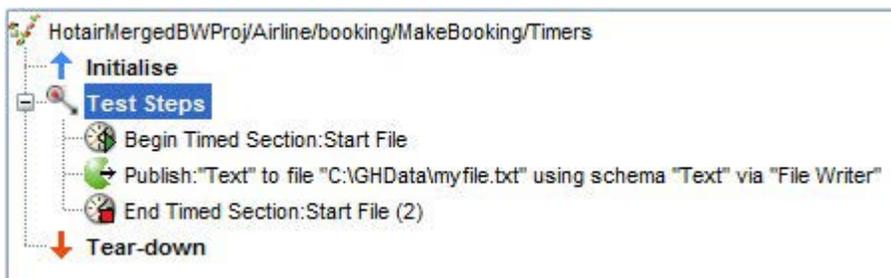
## 3.2 Creating a Test

In the following example, you will create a simple test that includes a timed section that contains a single Publish action. The Publish action will utilize the File transport to write text to a file (although any transport can be used).

**NOTE:** For more information about the details behind the following steps (for example, launching Rational Performance Test Server, creating resources, and so on), please see *IBM Rational Integration Tester Getting Started Guide* or *IBM Rational Integration Tester Reference Guide*.

1. If not already running, launch Rational Performance Test Server and open the project that was created in the last chapter.
2. In the Logical View of Architecture School, create and configure a service component that contains one operation and the **File Contents** resource.
3. Open the Test Factory perspective (press F10) and create a new test in the new operation.
4. Add a Publish action to the Test Steps phase and configure it to use the File transport that you created to write some simple text to a file.
5. Add a Begin Timed Section action (**Performance > Begin Timed Section**) before the Publish action and configure it as described in [Begin Timed Section](#).
6. Add an End Timed Section action (**Performance > End Timed Section**) after the Publish action and configure it as described in [End Timed Section](#).

Your simple test should look something like this:



7. Click **Save** (or press **Ctrl + S**) to save the test and run it to ensure it executes successfully.

**NOTE:** The timed section actions will have no effect at this stage since the test is not running as part of a performance test. The test will be executed as if the timing markers had not been added.

# Creating Performance Tests

## **Contents**

**[Creating a Performance Test](#)**

**[Adding a Distributed Test](#)**

**[Selecting a Test Engine](#)**

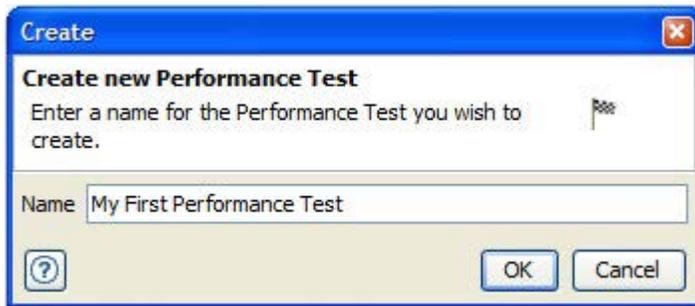
This chapter provides information about how to create a performance test that will utilize the simple test created previously.

---

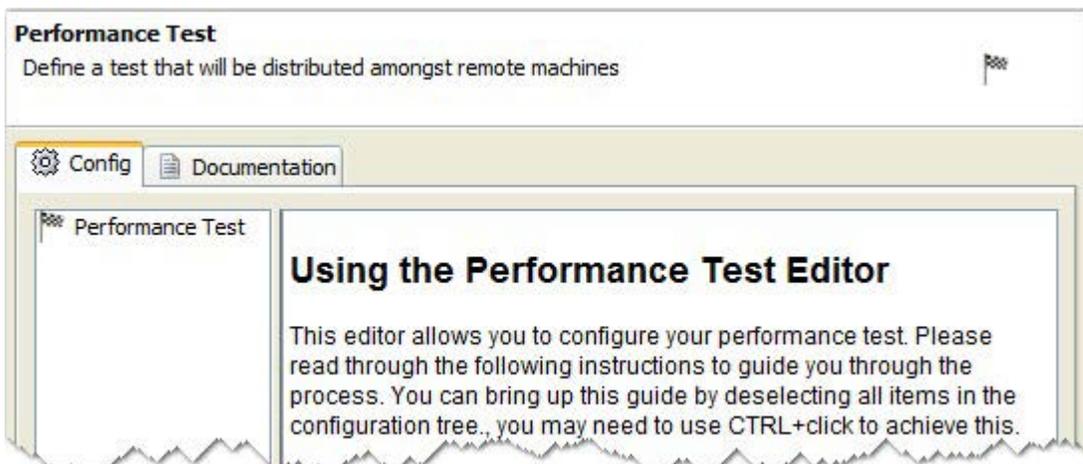
## 4.1 Creating a Performance Test

Now that you have created a simple test, you can create a performance test that will use it.

1. Open the Test Factory perspective (press F10).
2. Right-click the operation in the project resource tree and select **New > Tests > Performance Test**.
3. Provide a meaningful name for the test when prompted.



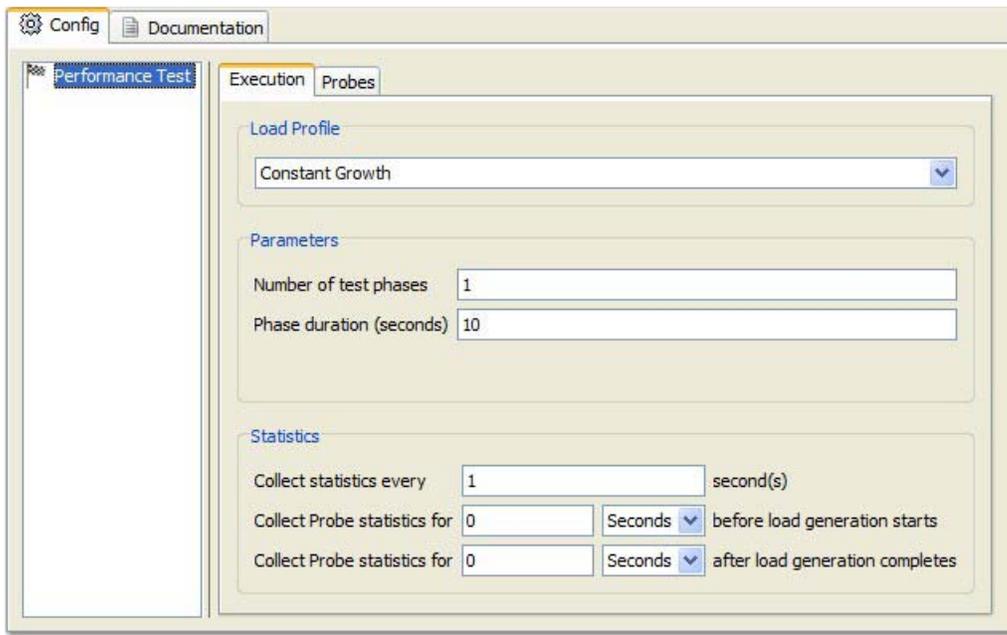
The new (empty) performance test is opened in the design canvas on the right side of the application.



The new test is displayed at the top of an empty tree under the **Config** panel. The tree is created as items are added to the test. When an item in the tree is selected, its details are displayed in the right-hand panel. With nothing selected (right-click the empty space beneath the tree), some documentation about using the test editor is displayed.

---

In our example, you will execute the test sequence 100 times to send 100 messages. This will be done by running it for 10 seconds with a message rate of 10 messages per second. We will not be altering the rate of messages so the test will have a single phase that lasts for 10 seconds.



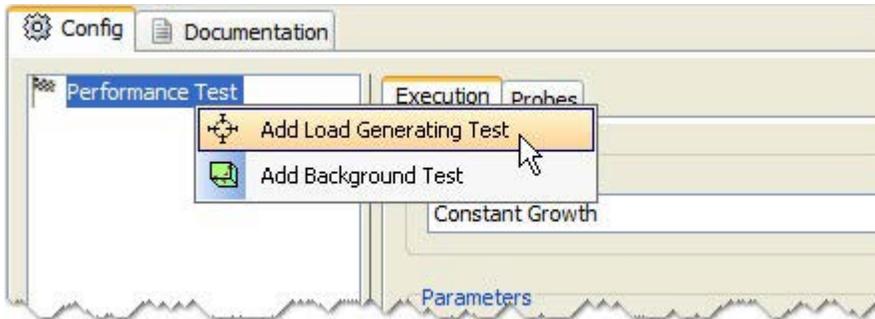
4. Enter **1** for the number of test phases and **10** (seconds) for the duration of the phase.
5. Enter **1** for the number of seconds between statistics intervals and **0** (seconds) for the next two statistics fields.

---

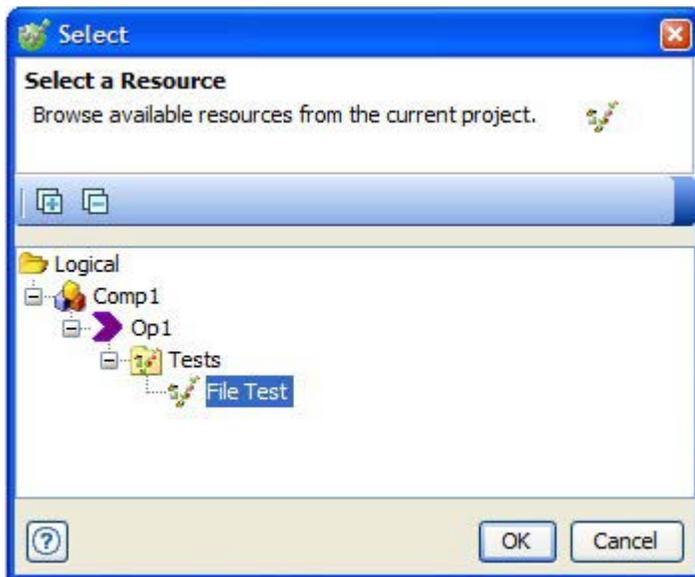
## 4.2 Adding a Distributed Test

Performance tests can execute any number of load generating or background tests, which are called distributed tests. In this section we will add a load generating test that runs the simple test that was created previously.

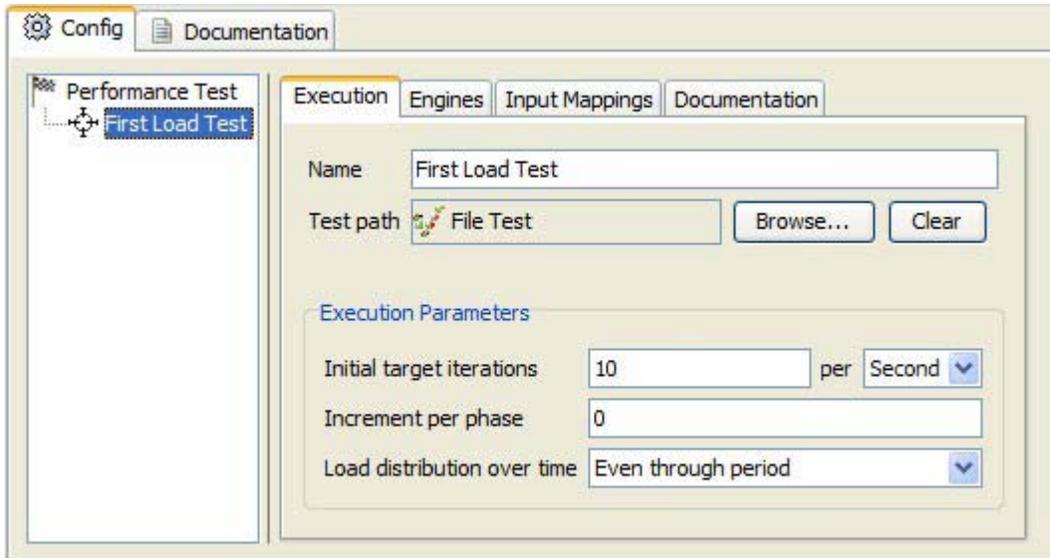
1. Right-click the root of the performance test and select **Add Load Generating Test**.



2. Enter a name for the test in the **Name** field.
3. Click **Browse** next to the **Test Path** field to locate and select the simple test that was created previously.



- 
- Under **Execution Parameters**, set **Initial target iterations (per second)** to **10** – this will generate 10 messages per second (that is, the test will run 10 times per each second and it sends one message each time it runs).



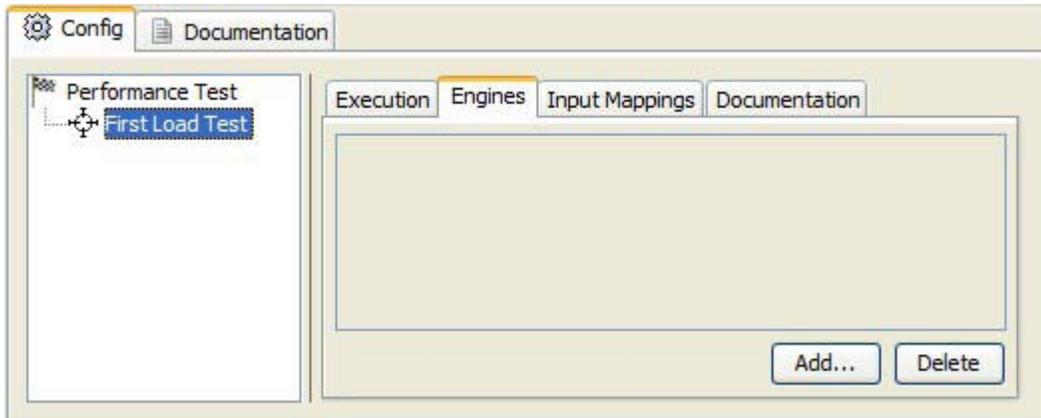
- The remaining fields can be left unchanged.

---

## 4.3 Selecting a Test Engine

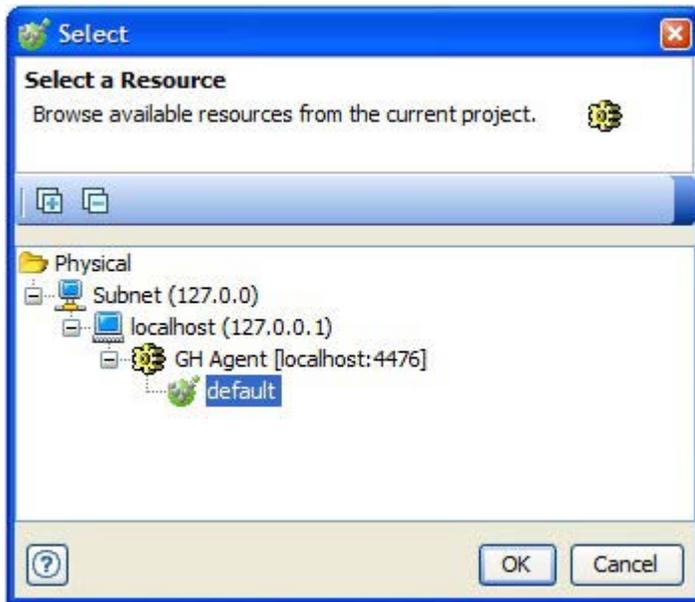
The final step in creating a basic performance test is to select an existing test engine that will execute the test.

1. Select the load generating test that was created in the previous section and click the **Engines** tab.



In a new test this tab will be empty, since no test engines have yet been configured.

2. Click **Add** to locate and select the default test engine that was created previously with the agent.



The **default** test engine should now be listed under the **Engines** tab.

# Running Performance Tests

## **Contents**

[Running the Agent](#)

[Executing the Test](#)

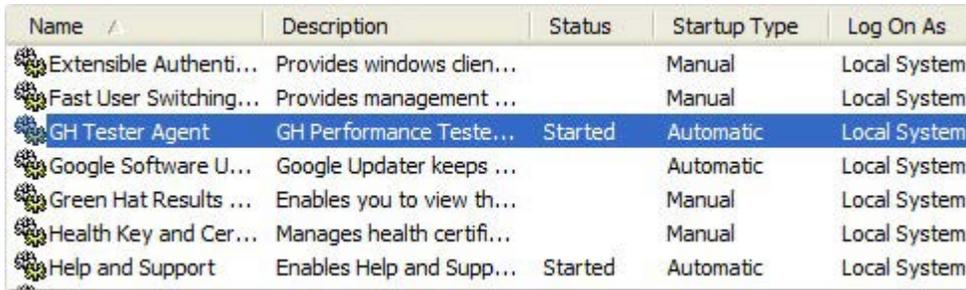
This chapter provides information about running the necessary agent and executing the performance test.

---

## 5.1 Running the Agent

Before you can execute your performance test, you must first verify that the agent (selected within the load generating test) is running.

Depending on how Rational Performance Test Server was installed, the agent may already be running as a Windows service.



Name	Description	Status	Startup Type	Log On As
Extensible Authent...	Provides windows dien...		Manual	Local System
Fast User Switching...	Provides management ...		Manual	Local System
GH Tester Agent	GH Performance Teste...	Started	Automatic	Local System
Google Software U...	Google Updater keeps ...		Automatic	Local System
Green Hat Results ...	Enables you to view th...		Manual	Local System
Health Key and Cer...	Manages health certifi...		Manual	Local System
Help and Support	Enables Help and Supp...	Started	Automatic	Local System

If not installed as a service, you can start the agent by launching either the desktop or Start menu shortcuts that can be installed (optionally).

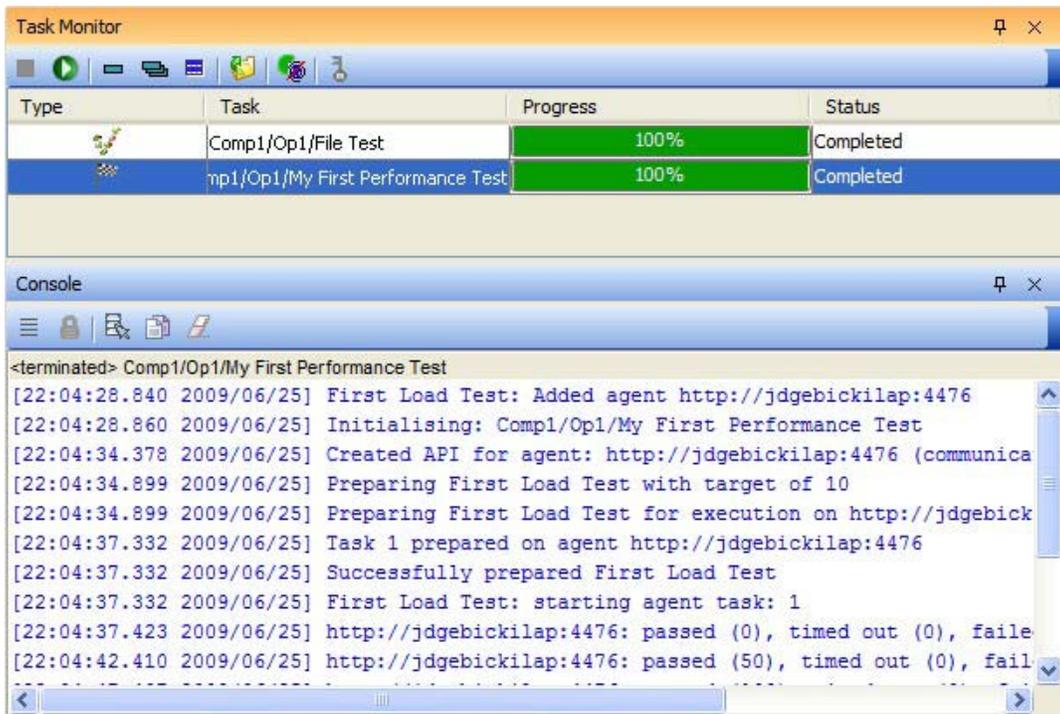
If necessary, you can start the agent by running **GHTesterAgent.bat**, located in the bin directory of the Rational Performance Test Server installation.

---

## 5.2 Executing the Test

Performance tests can be executed in the same way as other Rational Integration Tester resources. In this case, you will be using the Test Lab perspective.

1. Open the Test Lab perspective (press F11).
2. Select the performance test in the project resource tree and click **Run** (or press **F5**) in the main toolbar (you can also right-click the performance test and select **Run**).



The test should pass, as expected, and the results will be stored in the project database.

# Viewing Results

## **Contents**

**[The Results Gallery](#)**

**[Plotting Multiple Tests](#)**

**[Saving and Loading Charts](#)**

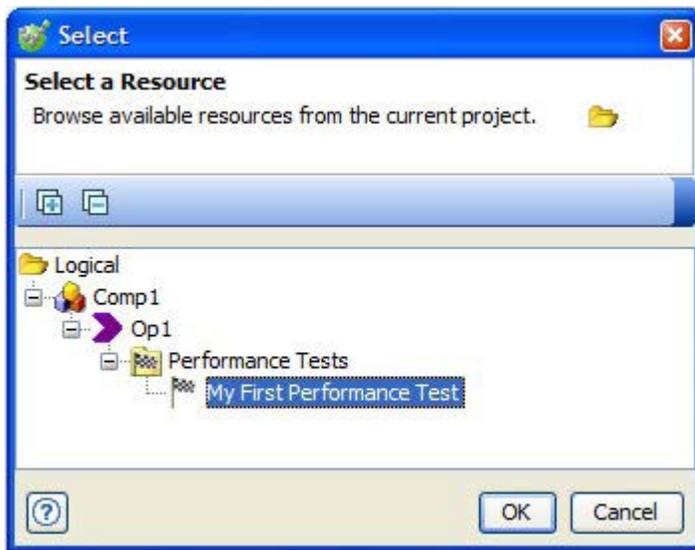
This chapter provides information about how to chart the results of your performance test in the Results Gallery perspective.

---

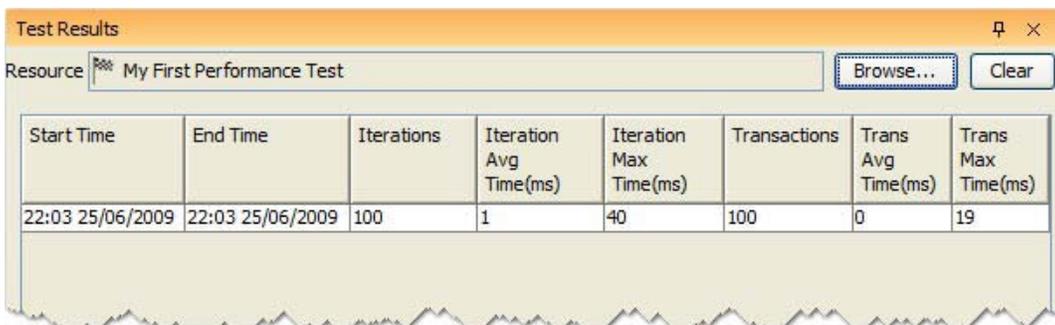
## 6.1 The Results Gallery

The results of your performance tests can be viewed in charts, available in Rational Performance Test Server's Results Gallery perspective.

1. Open the Results Gallery perspective (press F12).
2. Click **Browse** next to the **Resource** field to locate and select your performance test.



The execution results for the selected test (only one in this case) are displayed in the table (each line represents a data set for the selected performance test).

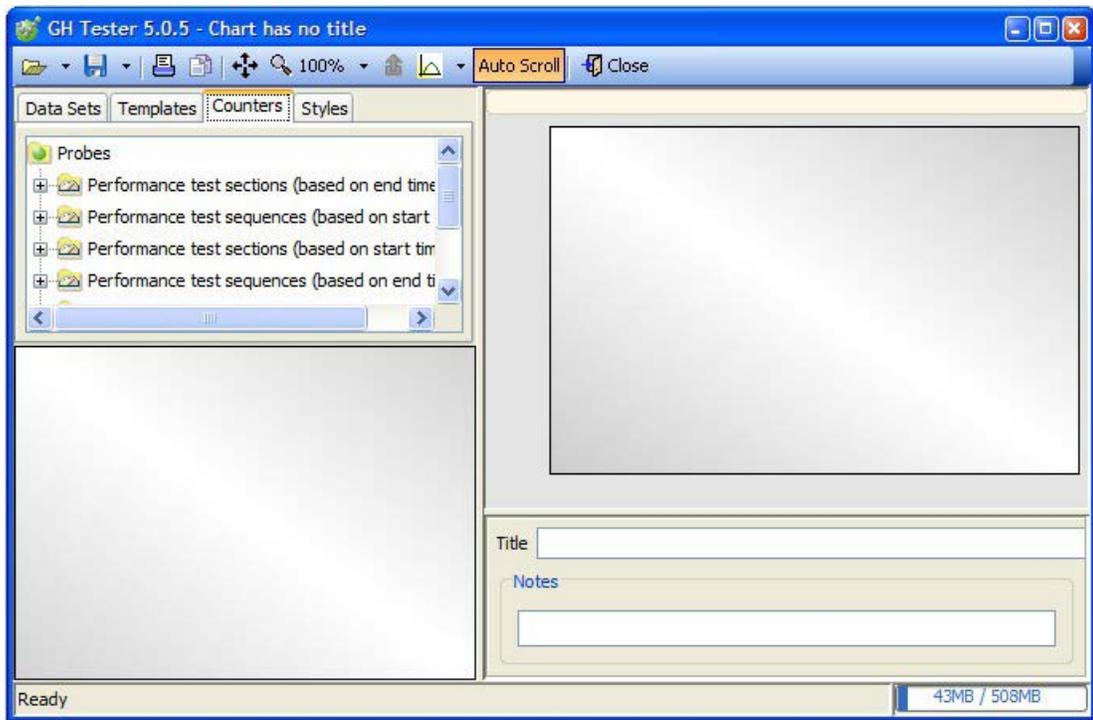


Start Time	End Time	Iterations	Iteration Avg Time(ms)	Iteration Max Time(ms)	Transactions	Trans Avg Time(ms)	Trans Max Time(ms)
22:03 25/06/2009	22:03 25/06/2009	100	1	40	100	0	19

3. Double-click the data set, or select it and click **Analyse Results**.

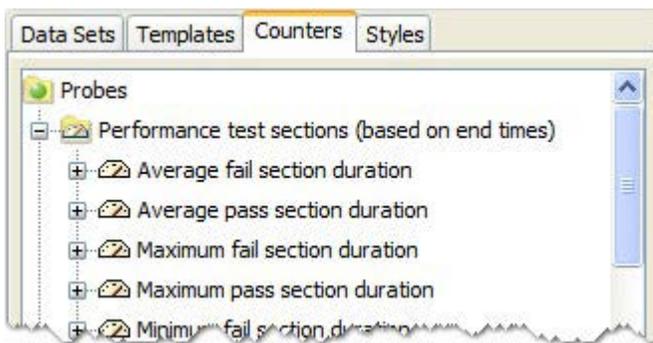
---

The database will be queried and the results of the selected data set will be opened in an empty chart window.



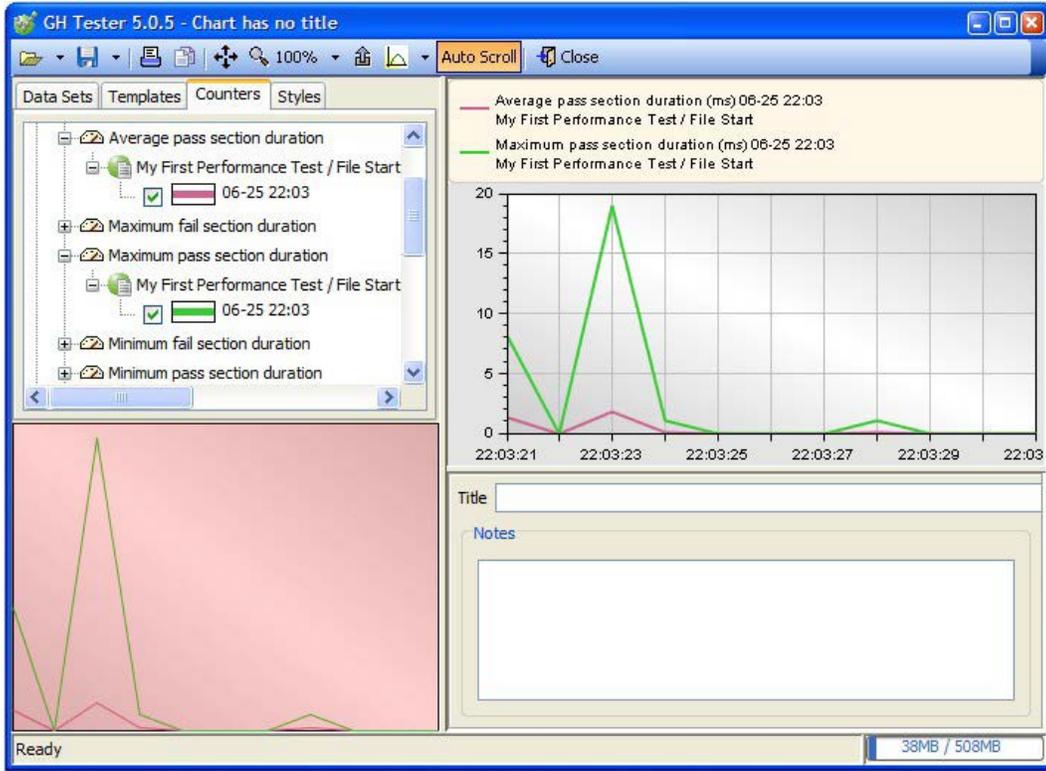
Under the **Counters** tab, the folders contain the sets of counters that were produced by the test run.

4. Double-click on **Performance test sections (based on end times)** to show the available counters that may be plotted.

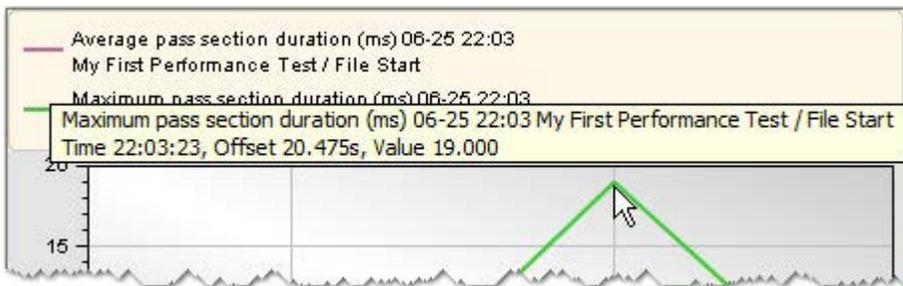


Expand the counters for Average pass section duration and Maximum pass section duration, then expand the test/timer entry within each one.

Tick the check box next to the date and time of the selected test run, and the data for each counter is immediately added to the chart.

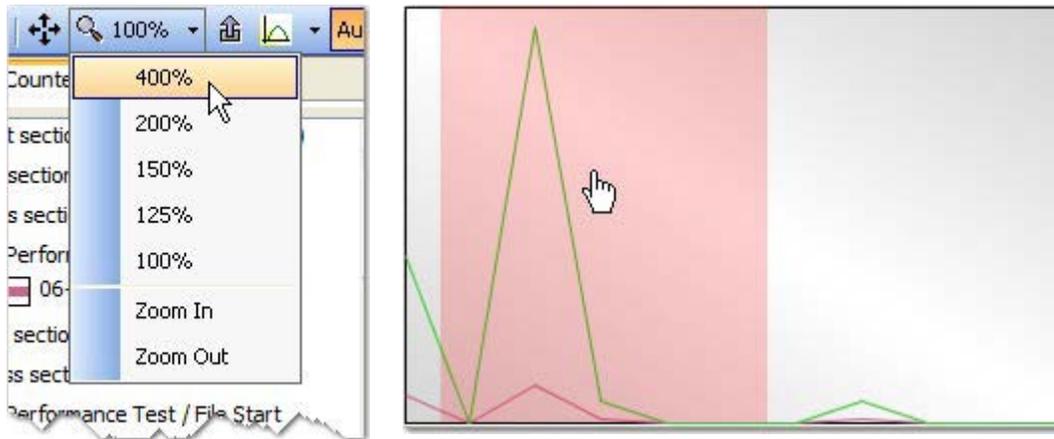


If you rest the cursor over the data points in the plot, a tooltip will display the details of each point.



---

The chart below the counters is a summary chart, which provides an overview of the entire chart. When the main chart is zoomed beyond 100% (using the **Zoom** menu), the shaded area in this summary chart can be dragged left or right to display a different portion of the main chart (that is, to display a smaller time interval).



You can add a title to the chart in the **Title** field. As you enter it, the title of the main window will also change.

To see more detail in the chart, you can maximise it within the chart window. by clicking the  icon to toggle full screen/normal view. When the chart is maximised, the configuration tabs, the summary chart, and the title and notes area are all hidden.

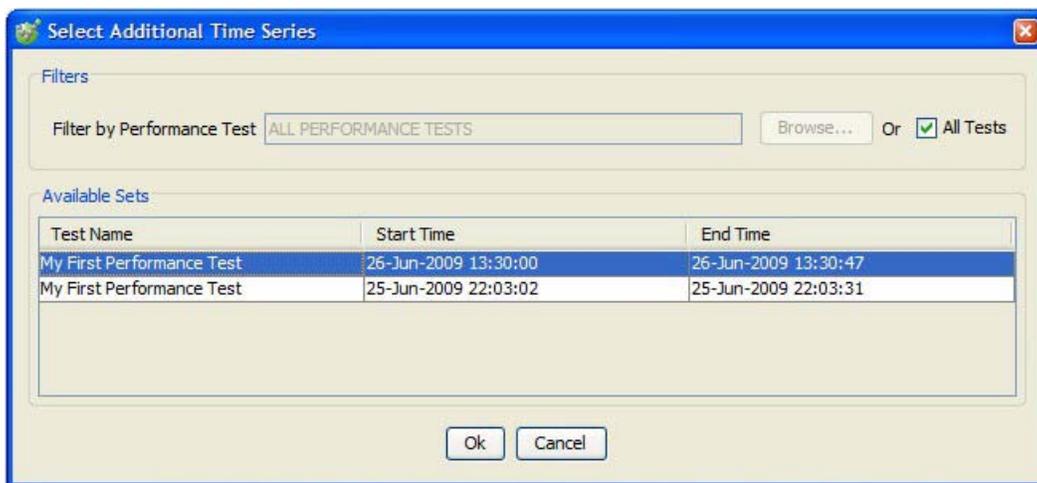
---

## 6.2 Plotting Multiple Tests

It may be useful to view multiple data sets in the same chart to compare successive test runs.

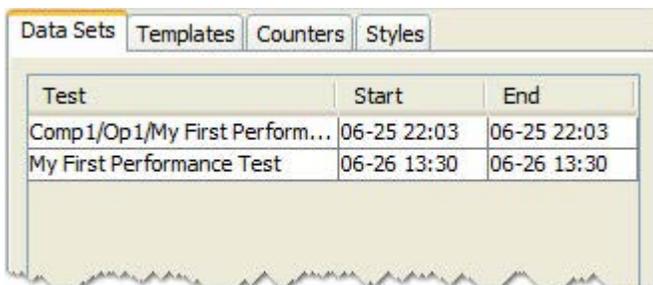
**NOTE:** To illustrate this, you will need to run the performance test at least one more time to generate additional data. You can minimize the chart window and run the test again in the Test Lab perspective. When finished, return to the chart window.

1. In the chart window, select the **Data Sets** tab.
2. Click **Add** to display the **Select Additional Time Series** window.

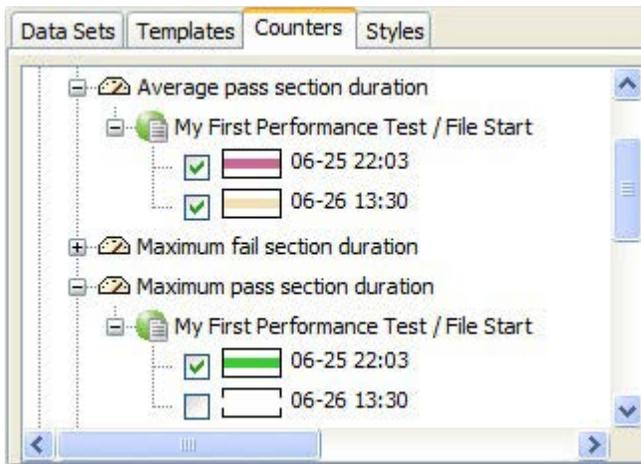


3. Enable the **All Tests** option, select another data set, and click **OK**.

The main window will show both test runs as individual data sets, ready for plotting.



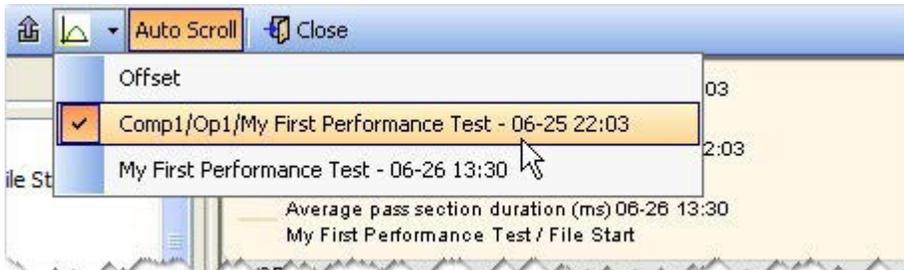
- 
4. Select the **Counters** tab and you will see additional test entries in all of the counters.



5. Select additional counters to plot on the chart.

The color of each plot line is selected randomly by the application, but you can change them by double-clicking on the colour bar in the Counters tree.

6. Use the **Offset** menu to select the times to display along the X-axis of the chart.



Selecting one of the test runs will display the actual times for that data set. If you select **Offset**, the elapsed time for all data sets is displayed, starting at zero.

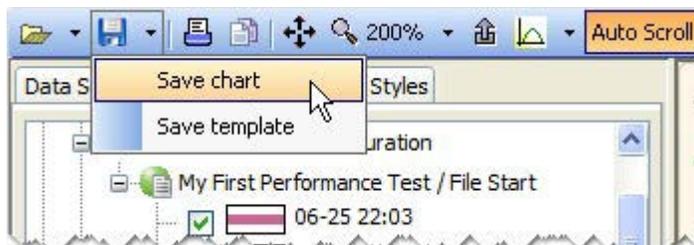
---

## 6.3 Saving and Loading Charts

Charts can be saved once they have been provided with a title, and these saved charts can then be loaded later on. Additionally, a chart can be saved as a template. The template will save the counter configuration, but the counters will be used to display the current data set. A saved chart, however, loads the actual chart style and data that were saved.

### Saving Charts and Templates

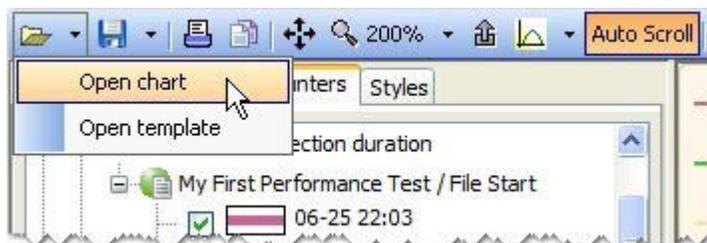
1. To save the current chart as a chart or template, click the arrow next to the **Save** icon and select the desired option.



2. If saving a template, select the location within the project tree and provide a name for the template. Charts will be saved using the chart title.

### Loading Charts and Templates

1. To load a chart or template, click the arrow next to the **Open** icon and select the desired option.



2. For charts, browse to the desired data set or enable the All Sets option, then select the image that represents the desired chart.
3. For templates, browse to the location where the template was saved and select the desired template.

**NOTE:** You can also apply saved templates from the **Templates** tab.

# Adding Probes

## **Contents**

[Adding a Probe](#)

[Configuring the Agent](#)

[Modifying Statistics Settings](#)

Probes collect data from running systems. Typically, probes are used to measure the impact of running tests on the target system (for example, the CPU load of the messaging server that handles the Rational Performance Test Server messages).

This chapter provides information about how to add probes to a performance test and configure how they are used.

---

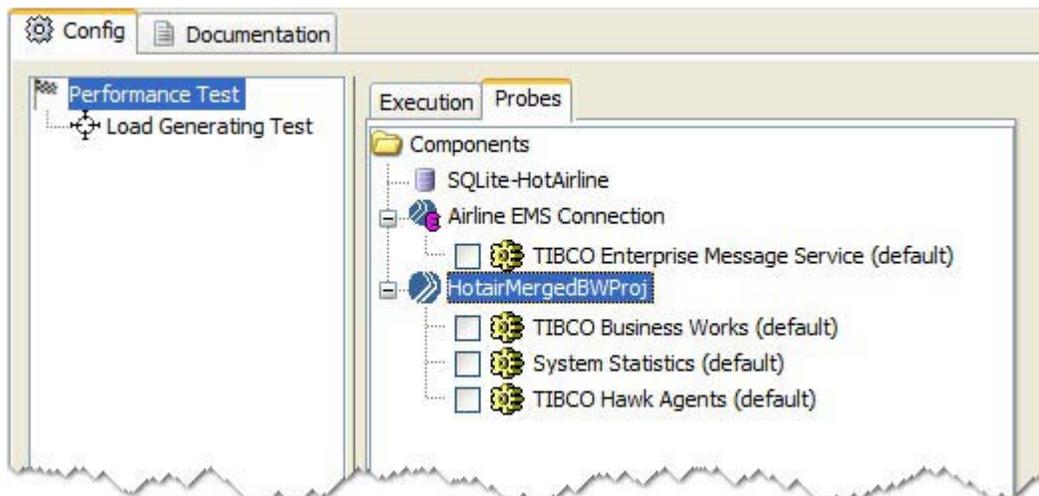
## 7.1 Adding a Probe

This section describes how to add a probe that will measure system statistics on the local machine that is running our tests. This is called the System Statistics Probe (Systat, for short).

**NOTE:** For the purpose of this example, a different Rational Performance Test Server project is being used, and the load generating test now points to a test that uses a Send Request/Receive Reply action pair, which utilize the TIBCO EMS transport. This is necessary to illustrate how to add probes to a performance test.

1. Open the performance test in the Test Factory perspective (press F10).
2. Select the root of the performance test and click the **Probes** tab.

A number of different probes are available for the selected test based on the transports and components that exist in the distributed test that has been selected in the performance test.



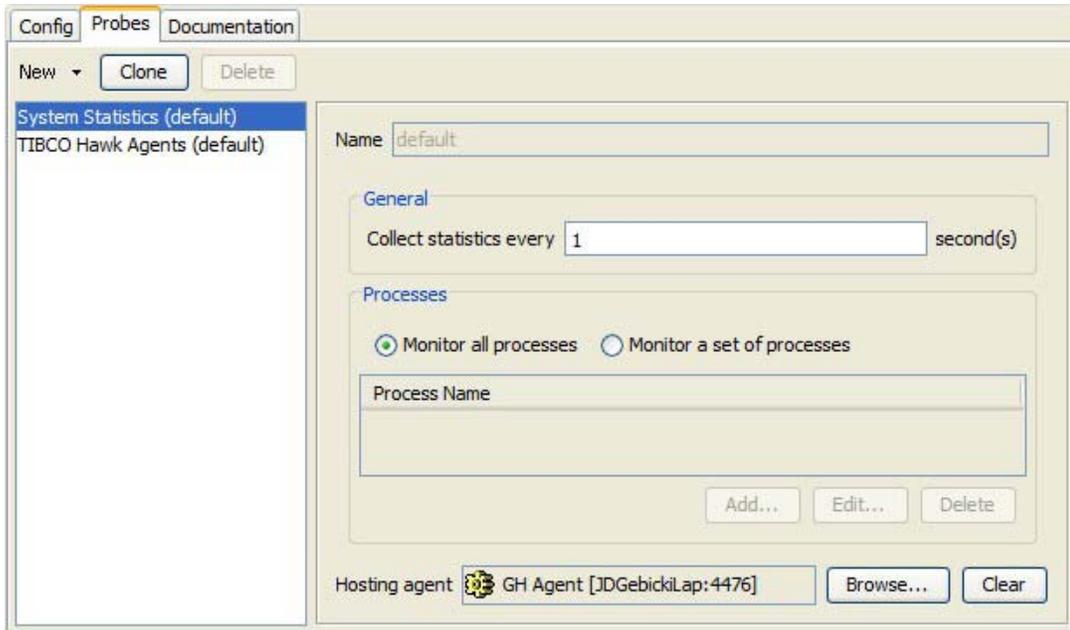
3. Tick the box next to the probes you want to add to the performance test (under the appropriate components).

---

## 7.2 Configuring the Agent

Once the probe has been added to the performance test, the probe settings and the agent that will run the probe must be verified/configured.

1. Open the Architecture School perspective (**F7**) and select the Physical View.
2. Double-click the host where your agent is installed and select the **Probes** tab.



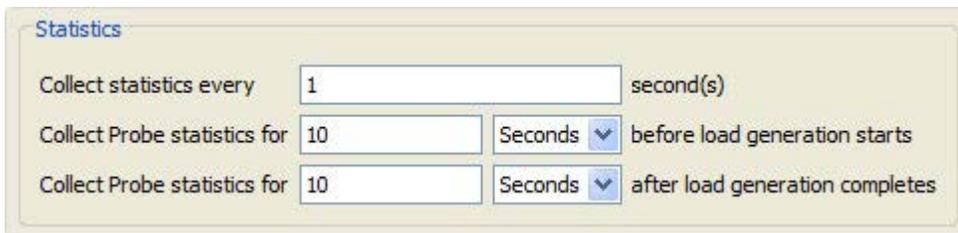
3. Ensure that the statistics interval is set to **1** second and the **Monitor all processes** option is selected.
4. Click **Browse** to locate and select the agent that will run the probe.

---

## 7.3 Modifying Statistics Settings

Now that a probe will be collecting statistics for the performance test, you can capture results before and after the actual test execution. This provides a more comprehensive view of the system in its idle state before the test and its state after the test has completed. For example, sending a large quantity of messages may keep the receiving servers busy after the test finishes as they process the data.

1. Return to the Test Factory perspective (press F10) to modify the execution options of the performance test.
2. Select the root of the performance test, select the **Execution** tab, and change the “before” and “after” entries for statistics to **10** (seconds).

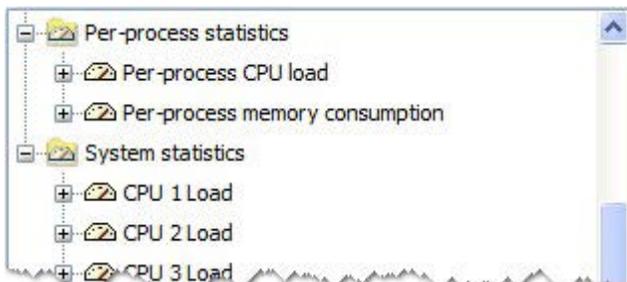


3. Save the changes and run the performance test again.

The test console will display considerably more output as probe status messages are added.

4. When the test finishes, open the Results Gallery perspective (press F12) and select the new run entry to analyze.

You will see that some new counters are available that were captured by the Sysstat Probe. The **Per-process statistics** counters contain entries for every process that was running during the test. The **System statistics** counters contains system measurements, such as memory usage, disk usage, and CPU load for up to 8 CPUs. The Total CPU Load counter shows a summary of CPU usage per machine.



The new counters can now be plotted along with the performance counters.

# Additional Options

## **Contents**

**[Mapping Tag Values with Data Sets](#)**

**[Chart Options](#)**

**[Configuring Test Loads](#)**

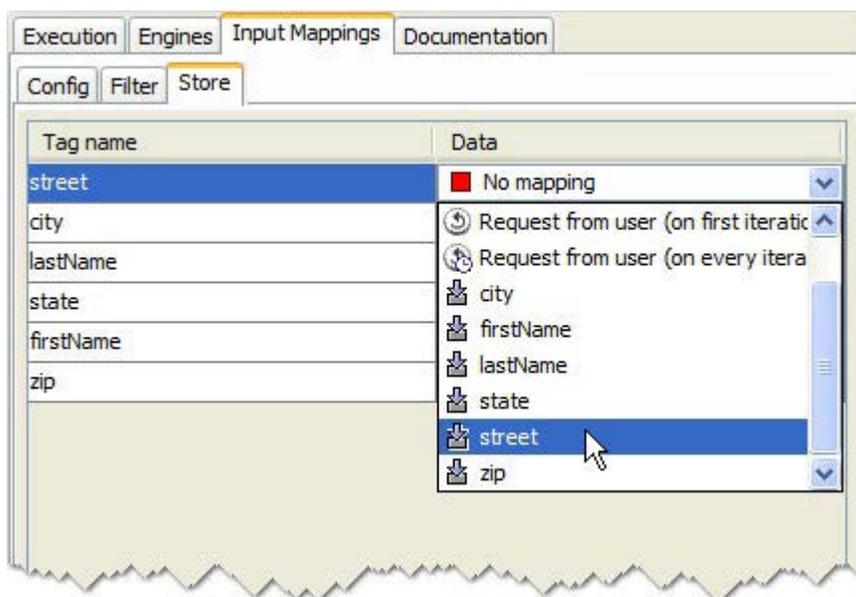
This chapter provides information about additional and more advanced settings and options that can be configured for performance tests and charts.

---

## 8.1 Mapping Tag Values with Data Sets

The **Input Mappings** tab of the performance test configuration allows you to select a data set and map the items it contains to tags contained in the distributed test.

1. Create a sample data source (in our example we are using a .csv file with customer data in it).
2. Edit the data source and copy the column names to the clipboard.
3. Edit the test used as your distributed test and open the tag data store.
4. Paste the copied column names in the data store to create tags from them.
5. Save the data source and distributed test.
6. Open the performance test, select the load generating test, and click the **Input Mappings** tab.
7. Under the **Config** tab, click **Browse** to locate and select the new data source.
8. Under the **Store** tab, select the desired options for mapping the data to the tags.



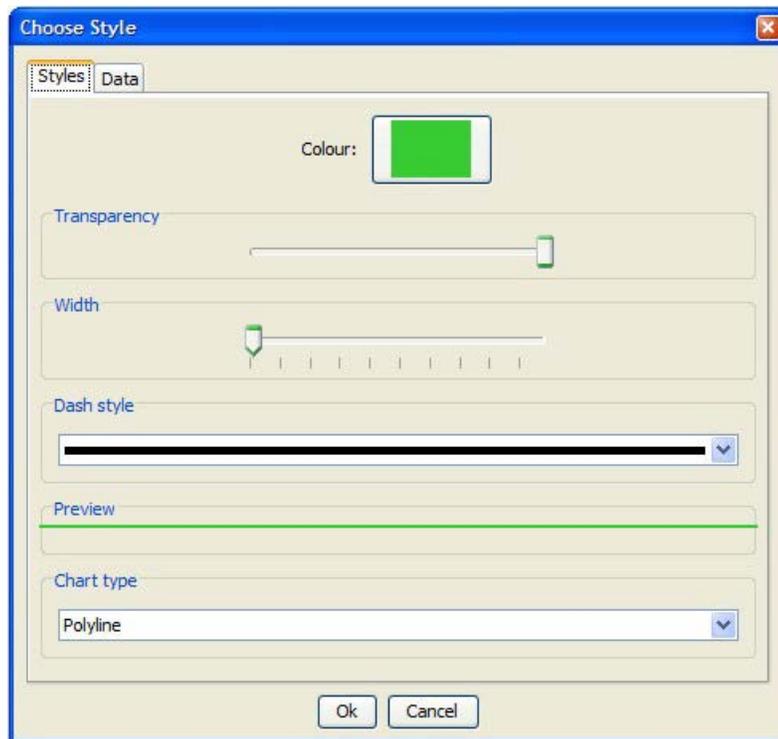
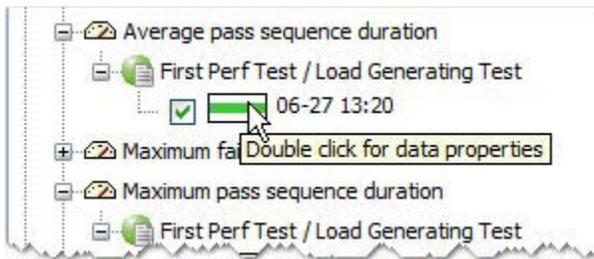
**NOTE:** If you add or delete tags from the distributed test, the changes will be reflected in the Input Mappings section, but the performance test will need to be re-saved.

---

## 8.2 Chart Options

In all of our examples so far we have viewed only a single chart and we have not modified any of the plot options. In this section we will look at some of the ways in which the style and appearance of the charts can be changed.

1. Open one of your performance test charts and ensure that you have more than one counter plotted.
2. Double-click one of the colour bars in the counter tree to display the properties for the plotted line.

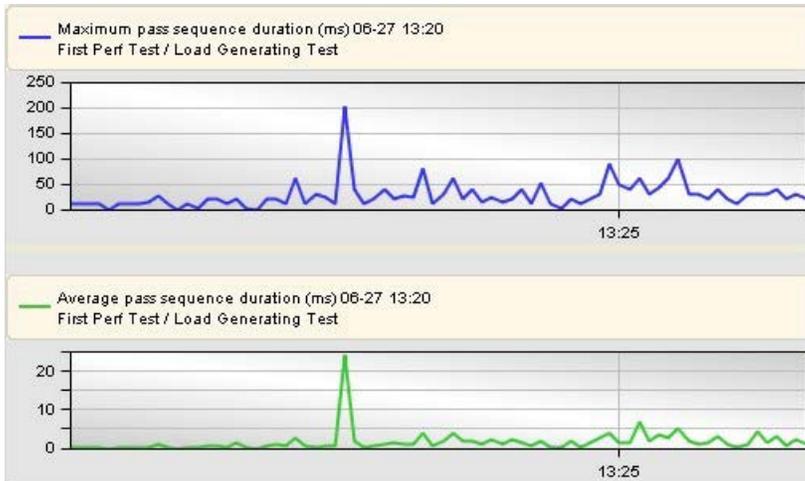


3. In the **Choose Style** dialog you can choose another colour by clicking the **Colour** block, and you can also change the line thickness and style.

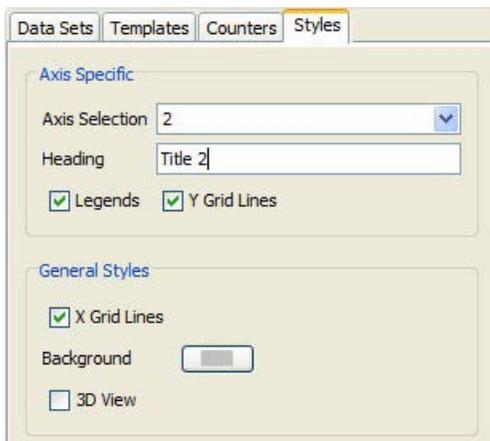
- 
- To plot the line on a new chart axis, select the Data tab and select **2** from the Axis combo box.

**NOTE:** Now that two axes are used, axis 3 will be available the next time you configure that properties of a plotted line.

Click OK to save the changes and the two charts will appear one above the other.



Under the **Styles** tab in the main chart window, you can assign a title to each axis, and you can customize the way grid lines and other items are displayed.



---

## 8.3 Configuring Test Loads

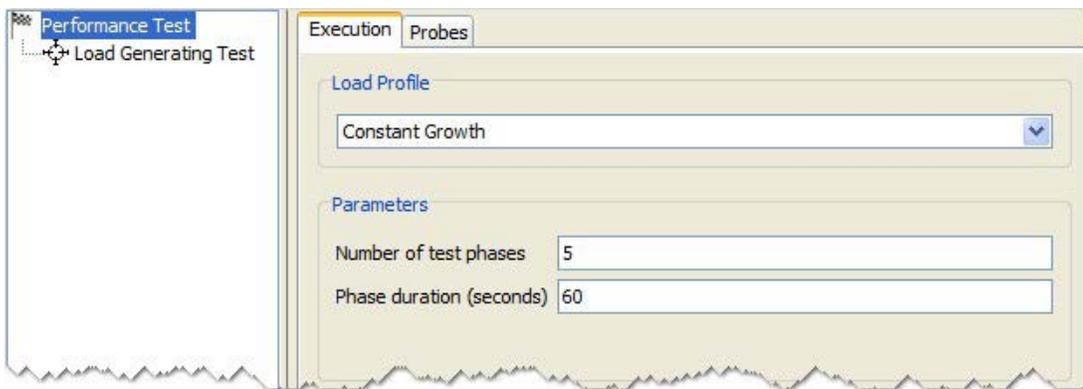
Performance tests are configured with a load profile that determines the way in which the tests in the performance test will run (that is, the rate at which new test iterations will be started). Two different profiles are available – **Constant Growth** and **Externally Defined**.

### 8.3.1 Constant Growth

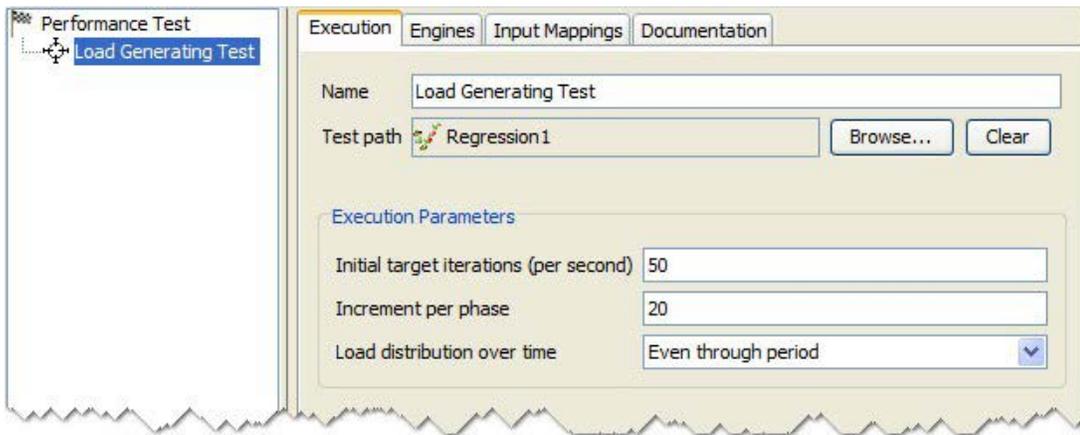
The **Constant Growth** profile starts tests at a given rate and increases the rate by a fixed amount for each new test phase. The rates are defined within the details of the individual tests. The parameters available for constant growth are the **Number of test phases** (minimum of 1) and the **Phase duration**, or the number of seconds each test phase should last.

A typical performance test will increase messaging load in a pre-determined way while measuring the impact on the receiving systems. In this example we will modify the test to initially send 50 messages per second, increasing by 20 in four successive steps. The test will therefore have five phases.

1. Configure five phases and increase the duration of each to 60 seconds.



2. Configure the initial message rate at 50 with an increment of 20.



3. Run the test – it will take five minutes to run – and examine the results

The chart shows messaging levels during each phase at 50, 70 and so on, as set in the configuration.



### 8.3.2 Externally Defined

When using the **Externally Defined** profile, the load is extracted from a test data set that has been previously configured within the project. This enables greater flexibility because the duration of each test phase can be different, as can the test rate.

The data set must contain two columns that will be used by the profile. The first column defines the phase duration and the second defines the target number of iterations.

---

In this example, we will create a data set containing three entries for three phases, each with differing message rates.

1. Create a comma-separated text file with the following data:

```
myTime,myIterations
30,50
60,75
120,100
```

2. Save the file as **phase.csv** on your machine.
3. Create a File Data Source from it called **Phases**.

**File Data Source**  
Configure this simple data source to use data from flat files as tag values in tests.

File Name: C:\GHData\phase.csv Browse...

Delimited  Fixed Width

**Format Configuration**

A row of the file contains column names

Rows to skip before column names: 0

Rows to skip after column names: 0

**Delimiter Options**

Comma  Tab  Space  Semi-colon  Other

Ignore delimiters within quoted strings Column count:  Calculate

Allow null values

Loop Data

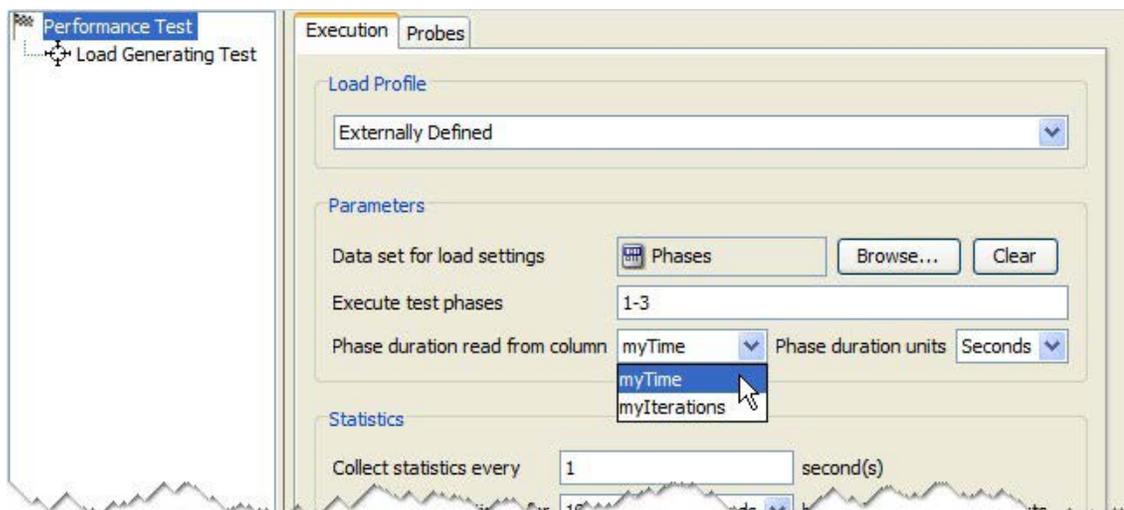
**Preview (max 4 rows)**

myTime	myIterations
30	50
60	75
120	100

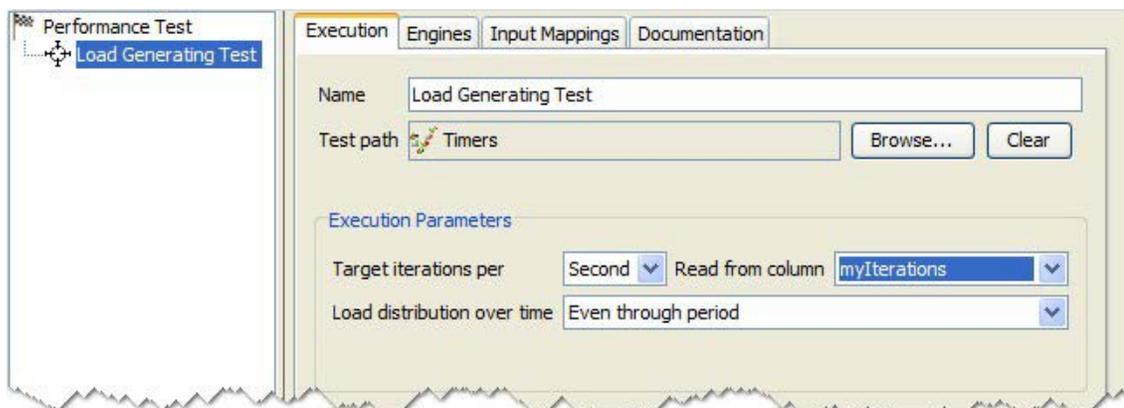
Refresh Copy column names to clipboard

4. In the performance test, set the load profile to **Externally Defined**.

5. Select the **Phases** data set under **Parameters**, enter “1-3” as the test phases to execute, and select the “myTime” column to read the phase duration.



6. Select the load generating test and set the execution iterations per second to the “myIterations” column of the data set.



7. Run the test and examine the results.

---

# Glossary

The following table below lists some of the key terms used in this document, and provides a description of each.

---

<b>Term</b>	<b>Description</b>
Agent	A special Rational Integration Tester process running on a host that allows test engine instances and probes to be launched on demand.
Background Test	A test executed on one or more test engines at a constant load level for the duration of the performance test.
Counter	An individual measurement from part of the system, examples include messages per second and CPU utilization.
Load Generating Test	A test that is executed by one or more test engines which may have varying load characteristics.
Host	The computer on which a software process runs.
JMS	Java Message Service, a J2EE technology. Several implementations of JMS exist, for instance IBM WebSphere® MQ, TIBCO EMS, and SonicMQ.
Performance Test Controller	Process that deploys probe and test configuration and orchestrates the performance test during execution. Communicates with Agents to achieve its objectives.
Probe	Measures information from part of the system and exposes it as one or more counters.
Result Set	The results of a performance test execution. One of these is generated every time a performance test is executed.
Server	A host computer on a network shared by more than one user.
Test Engine Instance	An instance of the Rational Integration Tester test engine, started by an agent, to execute a series of tests.

---

---

---

<b>Term</b>	<b>Description</b>
Transport	Informally, the messaging software in use (for example, JMS, TIBCO Rendezvous, TIBCO ActiveEnterprise, IBM WebSphere MQ, and so on).

---

---

# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT,

---

MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM United Kingdom Limited  
Intellectual Property Law  
Hursley Park  
Winchester  
SO21 2JN  
Hampshire  
United Kingdom

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the

---

capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corporation 2001, 2012.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

---

## Trademarks and service marks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

**IBM**<sup>®</sup>