

Executing Performance Tests

User's Guide



Version 5.0

Copyright

Copyright © 2004, 2005, 2006, 2007, 2008 Reflective Solutions Ltd.
All rights reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms and conditions of the Reflective Solutions Ltd. license agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine-readable form without prior consent, in writing, from Reflective Solutions Ltd.

Information in this document is subject to change without notice and does not represent a commitment on the part of Reflective Solutions Ltd.

THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER REFLECTIVE SOLUTIONS LTD. DOES NOT WARRANT, GUARANTEE OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY OR OTHERWISE.

Trademarks or Service Marks

jTune™ and StressTester™ are registered trademarks of Reflective Solutions Ltd.

All other trademarks are the property of their respective companies.

Contents

Introduction	4
Designing a Performance Test.....	5
Test Goal.....	5
Information Requirements	5
Injection Locations.....	6
User Journeys	6
Resource Monitors	6
Location Load Profiles.....	7
Considering Injection Locations	8
Run Components Workspace	9
Run Components Navigation Tree	9
Right-Click Menus	10
Injectors	11
Creating an Injector Configuration.....	11
Advanced Settings	13
SSL.....	13
Network Settings.....	14
Process Settings.....	14
Debug	16
Creating Test Run Configurations.....	17
User Journey Configuration.....	18
Resource Monitor Configuration.....	19
Starting Run Components.....	20
Starting Injectors	20
Starting Monitor Agents.....	20
Executing a Performance Test.....	21

Introduction

This user guide details the steps that you should consider in order to design and execute a performance test using StressTester™.

The design of the performance test is the last stage in ensuring that the load created during the test will be realistic when compared to the real World.

The sections of this guide cover everything that you should consider when designing a test, and then how you can configure and execute the designed test using StressTester™.

Designing a Performance Test

The following sections detail the areas that you should consider when designing your performance test.

Test Goal

The key thing to consider before configuring and executing a performance test is:

What is the performance test's purpose?

Typical performance test purposes include the following:

Load Test: what response times will the users receive at a certain load and transaction mix?

Stress Test: what is the maximum number of users that can be supported before the system “breaks”?

In this scenario, break may refer to the application failing, an unacceptable percentage of requests returning errors, or unacceptable response times occurring.

Spike Test: how does the application respond to a sudden short burst of high load?

Stability Test: are response times and system resource usage the same if a steady load runs for a long period of time (for example, for 24 hours)?

Information Requirements

Once the purpose of the test is established, the main reason for executing a performance test – to obtain information about the performance profile of the application under test – can be addressed.

What is the information the tester/team/IT department/business requires from the performance test?

In what format should the information be provided (graphs, data tables or both)?

The list of potential recipients of the information is provided on purpose; StressTester™ can be used to provide information to many different audiences from the same performance test due to its flexible and customisable analysis capabilities.

The [Analysing Performance Test Results](#) user guide contains more details on these capabilities.

Injection Locations

The next consideration is to determine from where the load should be generated.

StressTester™ has been designed to support distributed load simulation from multiple locations with little extra configuration effort required.

This flexibility allows you to simulate the correct number of users, at the correct locations, running correct User Journeys – truly realistic testing.

User Journeys

An obvious design consideration is which User Journeys will run during the performance test.

In addition, if load is being simulated from multiple locations, you should consider which User Journeys will run at which locations.

Resource Monitors

Depending on the purpose and information requirements of the performance test, you may wish to monitor the system under test's resources in order that performance problems can be diagnosed.

If this is required, you need to consider the Resource Monitors required, and which aspects of the system they should monitor.

The [Monitoring Underlying System Resources](#) user guide contains more details on StressTester™'s Resource Monitor capabilities.

Location Load Profiles

The final consideration is: what load should initially be simulated, for each User Journey, at each location?

In addition, if you intend to execute a stress test: how should these loads increase during the performance test?

Once again, it is important that this information is as correct as possible to ensure the performance test is realistic and the information obtained of use to the intended audience.

Considering Injection Locations

The StressTester™ component that simulates users interacting with the application under test is called an “Injector”.

Injectors can be thought of as synonymous with locations – typically one Injector will be installed at each location from which load will be simulated.

The granularity of “location” is dependent on the requirements of the performance test.

It could be that an Injector is installed on each floor of an office building to check the network components on each floor can handle that floor’s load.

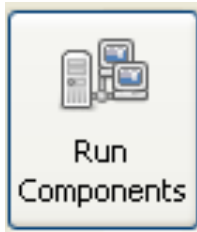
Alternatively, locations may be of a courser granularity, for example: New York, Chicago, San Francisco and London.

At each location identified, the Injector software must be installed and configured.

The [Installation](#) user guide provides details of how to install Injectors.

Run Components Workspace

All tasks related to configuring the components used to execute a performance test, and creating the test profile – known as the Test Run Configuration – are defined in the Run Components workspace.



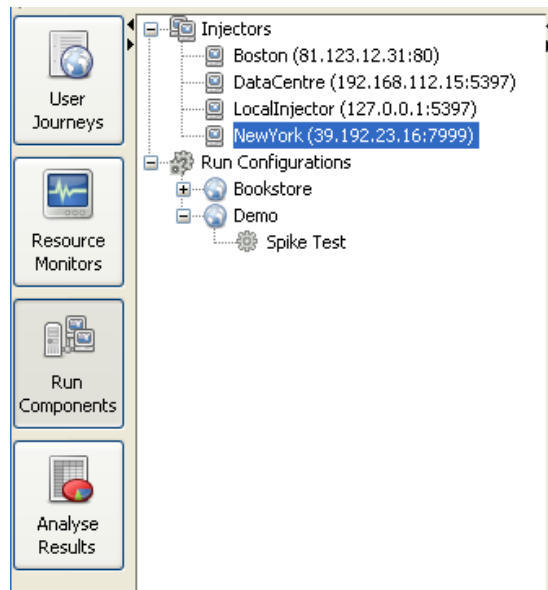
To access this workspace, simply click the “Run Components” button on the left hand side of the StressTester™ user interface.


Run Components Navigation Tree

When the workspace is entered, the workspace navigation tree and introductory help page are displayed.

Note: It is highly recommended that you read the introductory help page the first time you use the workspace – it contains valuable information.

By clicking a node in the navigation tree, the introductory page will be changed to a data panel - allowing the selected node's information to be viewed and updated (StressTester™ license permitting).



In the screen shot example shown, there are four Injectors (represented by the  icon) – indicating the different locations from which load can be generated.

In addition, each application has a node under the Run Configurations node, and by expanding a node,

you can see the Test Run Configurations that are defined for the application (represented by the  icon).

Right-Click Menus

The majority of nodes within the Run Components workspace navigation tree have context sensitive right-click menus – providing easy access to the relevant StressTester™ functionality.

The sections below detail the different functionality available on the different tree nodes.

“Injectors” Node

Menu Item	Description
Create Injector	Creates a new Injector configuration.

Injector

Menu Item	Description
Delete	Delete's the Injector configuration.

Application

Menu Item	Description
Create Run Configuration	Creates a new Test Run Configuration.

Test Run Configuration

Menu Item	Description
Execute	Start a performance test as specified by the Test Run Configuration.
Delete	Deletes the Test Run Configuration and all results relating to it. Note: When you delete a Test Run Configuration you will no longer be able to view test results created by executing the configuration.

Injectors

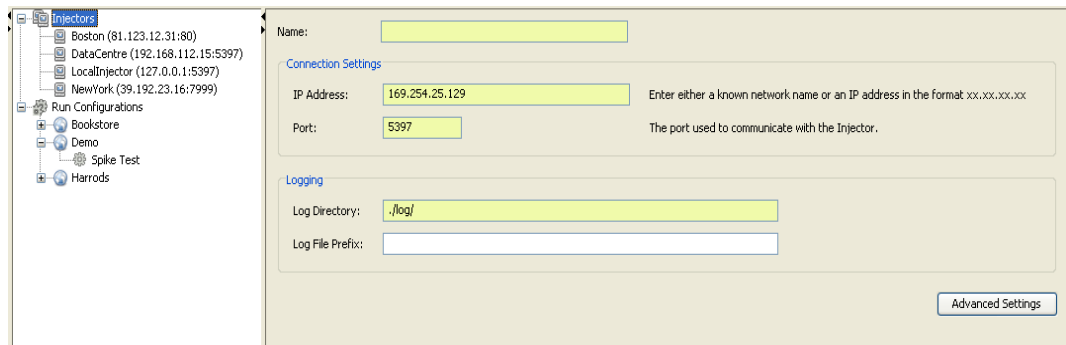
As stated above, an Injector is synonymous with a location from which simulated load should be created.

Creating an Injector Configuration

You only need to define an Injector once within StressTester™. After this, it can participate in as many Test Run Configurations as you like.

To define an Injector within StressTester™, right-click on the Injectors node and then select the “Create Injector” option.

A screen similar to that shown below will be displayed.



The fields on the screen are as follows:

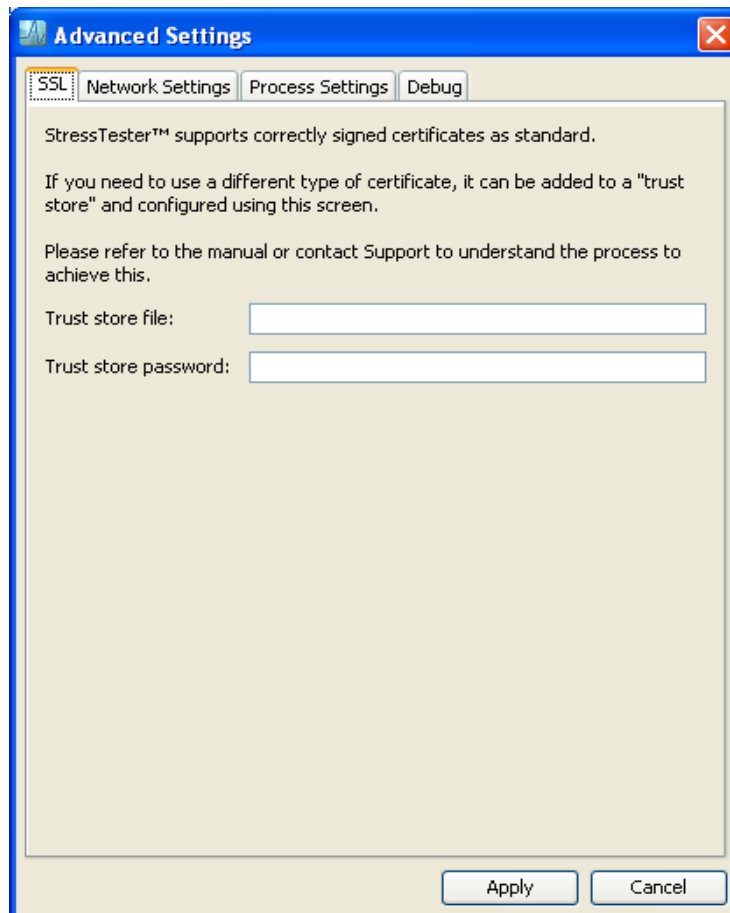
Field	Description
Name	A unique name to give to this Injector. It is recommended names reflect the location of the Injector.

Field	Description
IP Address	<p>The IP address (in the format xx.xx.xx.xx) of the machine hosting the Injector.</p> <p>By default, StressTester™ suggests the IP address of the machine you are working upon.</p>
Port	<p>A port that can be used to send commands to the Injector.</p> <p>By default StressTester™ suggests a port number of 5937 – this is the port reserved by the IANA¹ for StressTester™ Injectors.</p>
Log Directory	<p>The path of a directory (either relative to the starting location of the Injector, or an explicit path name) within which this Injector will write log and error files.</p>
Log File Prefix	<p>A prefix to be applied to all log and error files. This is useful if you configure two Injectors to write to the same log directory.</p>

¹ Internet Assigned Numbers Authority.

Advanced Settings

By clicking the “Advanced Settings” button on the Injector properties screen, you can access advanced properties.



The fields on the different tabs on this pop-up are described in the sections below.

SSL

Secure communications between the server and client require the client to trust the server to be what it claims to be.

This is achieved using server certificates that identify the server and are signed by a trusted third-party authority.

StressTester™ supports all certificates correctly signed by default.

However, if it is necessary to use a certificate that is not signed in such a way, this tab can be used to describe the file containing the certificate to be used, and the password used to read the file.

The table below describes the fields on this tab.

Field	Description
Trust store file	The file containing the certificate to be trusted by StressTester™.
Trust store password	The password used to access the file.

Please contact Reflective Solutions Support – support@reflective.com – for details on how to configure the “trust store”.

Network Settings

This tab allows you to configure details of a proxy that StressTester™ should use to access the application under test.

The table below describes the fields on this tab.

Field	Description
Proxy Address	The IP address of the proxy to use.
Proxy Port	The port on which the proxy ‘listens’.

Process Settings

StressTester™ Injectors are implemented in Java. This means an Injector runs on one or more Java Virtual Machines (JVMs).

By default, StressTester™ Injectors use the host machines default Java implementation, and a new Injector process is spawned for every 200 users simulated.

This tab allows these and other default parameters to be overridden.

Note: These are advanced parameters and should only be updated by experienced users with a detailed knowledge of JVMs, or when you are instructed to do so by Reflective Solutions Support.

The table below describes the fields on this tab.

Field	Description
Injector Process JVM	<p>The full path to the Java command that should be used to run the Injector process(es).</p> <p>By default, the host machine's default Java implementation will be used to run the Injector process(es).</p>
Supplementary Classpath	<p>Additional items that should be placed on the Injector's Java class path.</p>
Memory: Initially Assigned	<p>The amount of memory that the JVM should initially request from the operating system when the Injector starts.</p> <p>By default, Java's standard setting is used. This varies depending on the Java implementation.</p>
Memory: Maximum Limit	<p>The maximum amount of memory that the JVM running the Injector can request during a performance test.</p> <p>By default, Java's standard setting is used. This varies depending on the Java implementation.</p>
Supplementary Arguments	<p>Java runtime arguments that will be passed to the JVM when the Injector is started.</p>
Maximum Threads	<p>In order to avoid JVM threading issues, a StressTester™ Injector will create a clone of itself when it has allocated a set number of threads.</p> <p>In this context, a Java thread is synonymous with a simulated user.</p> <p>Adjusting this field changes when clone processes are created.</p> <p>Note: The original Injector process manages all its clones and ensures they are killed at the end of a performance test.</p>

Debug

Under the instruction of Reflective Solutions Support, you may be asked to configure debug flags in order that more information about a reported issue may be captured.

This is the screen where these flags are configured and it should only be used following guidance from Reflective Solutions Support.

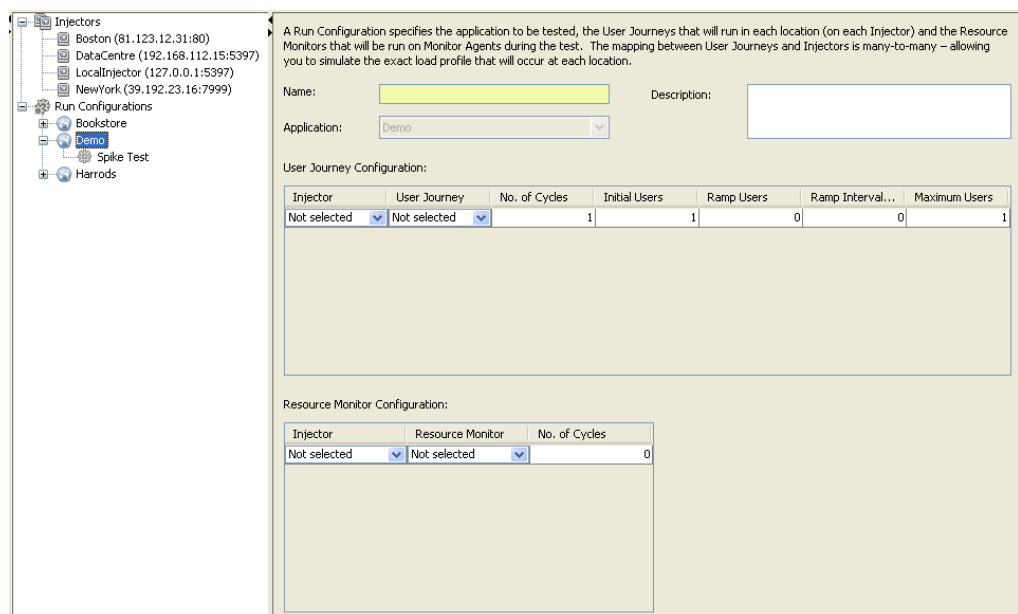
When you have updated the Injector properties, click the “Save” toolbar button to persist the changes.

Creating Test Run Configurations

When you have designed your performance test and configured the Injectors that will be used during the test, you can create a corresponding Test Run Configuration within StressTester™.

To create a new Test Run Configuration, right-click on the relevant application node and then select the “Create Run Configuration” option.

A screen similar to that shown below will be displayed.



The fields on the screen are as follows:

Field	Description
Name	A unique name for the Test Run Configuration.
Description	An optional free text description of the Test Run Configuration. It is recommended you use this field to record the purpose of the configuration i.e. the goal of the performance test.
Application	The application that will be tested by the Test Run Configuration. This field is read-only on this screen.

Field	Description
User Journey Configuration	Please see the section below for details of this table.
Resource Monitor Configuration	Please see the section below for details of this table.

User Journey Configuration

This table within the Test Run Configuration screen allows you to define which User Journeys will be run from which locations, and how the number of simulated users for a User Journey at a location will be ramped.

Note: The exact numbers of Injectors you can use, the number of users you can simulate, etc. may be restricted by your StressTester™ license.

The fields in the table are described below.

Field	Description
Injector	The Injector to use to simulate the User Journey.
User Journey	The User Journey to be executed on the Injector. Note: The relationship between Injector and User Journey is many-to-many: an Injector can run many different User Journeys, and a User Journey can run in many different locations.
No. of Cycles	The number of iterations of the User Journey each simulated user should complete during the performance test. A value of '0' indicates that the simulated users should continue to run until you stop the performance test.
Initial Users	The initial number of users that should be simulated for the selected User Journey running on the selected Injector.
Ramp Users	The number of simulated users that should be added to those already running every time the number of users is ramped.

Field	Description
Ramp Interval	The time period in seconds between subsequent additions of Ramp Users to those users already running.
Maximum Users	The maximum number of simulated users that should run the selected User Journey on the selected Injector. When this number of simulated users is reached, no further ramping occurs.

Resource Monitor Configuration

This table within the Test Run Configuration screen allows you to define which Resource Monitors will measure underlying system metrics during the performance test.

The fields in the table are described below.

Field	Description
Injector	The Injector to use to communicate with the Monitor Agent that is executing the Resource Monitor.
Resource Monitor	The Resource Monitor to run to measure a system metric during the performance test.
No. of Cycles	The number of samples of the metric to measure during the test. This is normally left to its default value of '0' – indicating that the metric should be measured for the duration of the performance test.

When you have updated the Test Run Configuration properties, click the “Save” toolbar button to persist the changes.

Note: For further details on the fields on this screen, press the Help button when on the screen to access the in-tool context specific help.

Starting Run Components

You are now ready to start the processes that are defined within the Test Run Configuration.

These processes are either Injectors or Monitor Agents.

Starting Injectors

To start an Injector, navigate to the directory in which StressTester™ is installed and then run the `injector.cmd` or `injector.sh` command (depending on your operating system) with a single parameter – the name of the Injector as it appears in the Injector configuration.

For example, to start an Injector that has been named “Boston” on a Microsoft Windows machine, you would run the command:

```
injector.cmd "Boston"
```

Note: Injectors, once started, are long running processes that will only stop when you kill the process. This means that once an Injector has been started once, it does not need to be started again until you either kill the Injector or restart the machine.

Starting Monitor Agents

To start a Monitor Agent (the process that executes Resource Monitors), navigate to the directory in which you installed the Monitor Agent and then run the `monitor_agent.bat` or `monitor_agent.sh` command (depending on your operating system).

The command does not require any parameters but the Monitor Agent properties should have been correctly configured prior to it being started.

Refer to the [Monitoring Underlying System Resources](#) user guide for details on how to install and configure Monitor Agents.

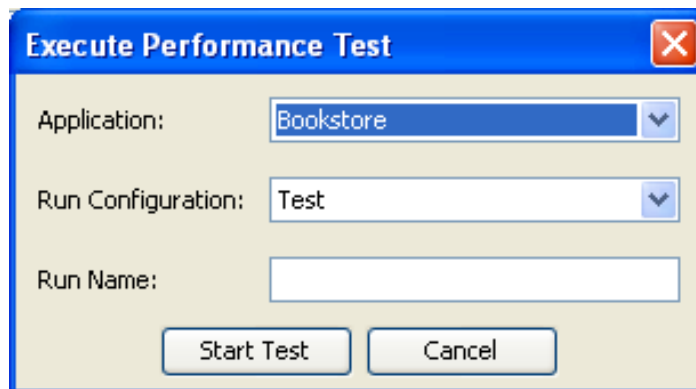
Note: Like Injectors, Monitor Agents are long running processes that will only stop when you kill the process or restart the machine.

Executing a Performance Test

Once you have defined a Test Run Configuration and started the components it requires, you can execute a performance test.

There are two ways to start a performance test: either right-click on a Test Run Configuration navigation tree node and choose the “Execute” option or click the “Execute Test Run” button on the StressTester™ user interface toolbar.

Whichever you choose, you will be presented with the following pop-up screen.



The fields on the screen are as follows:

Field	Description
Application	The name of the application to be tested. This field will be read-only if the execution was invoked by right-clicking the Test Run Configuration navigation tree node.
Run Configuration	The Test Run Configuration that defines the performance test to be executed. This field will be read-only if the execution was invoked by right-clicking the Test Run Configuration navigation tree node.
Run Name	A unique name for the performance test.

When you have completed the screen fields, click the “Start Test” button to start the performance test.

At this point, the StressTester™ user interface will perform the following:

- ✓ Communicate with all the run components involved in the performance test to ensure they are running and ready to execute the test.

If a component is not running or ready, you will see an appropriate error message and the performance test will not be executed.

- ✓ Instruct all the components of the performance test that will be executed – in order that the components can understand what they need to do in the performance test
- ✓ Switch the user interface display to the Analyse Results workspace with the current test selected in that workspace's navigation tree

To understand how to use the analysis screen you will see, and how to further configure the display and analyse the performance test results, please refer to the [Analysing Performance Test Results](#) user guide.