# 30 Minute Introduction to StressTester™


**Version 6.0**

# Copyright



# Restricted Rights Legend



# Trademarks or Service Marks

# Contents

# Introduction

The purpose of this guide is to allow you to understand the terminology and architecture of StressTester™; how to use the StressTester™ user interface; and guides you through recording a business transaction and executing your first test.

It is presumed that you have installed StressTester™, obtained a license and can successfully start the StressTester™ user interface.

Although the guide may seem to have many pages, this is because screen shots have been provided wherever possible to ensure you understand everything presented.

It is expected, that if the business transaction you choose takes no more than 5 minutes to record, to follow this guide in its entirety will require no more than 30 minutes of your time.

We would encourage you to make this investment; it will mean you have a good foundation of knowledge.

Other user guides in the StressTester™ documentation set (located within the `/doc` directory of the StressTester™ installation) are:

- Installation User Guide
- Recording HTTP User Journeys
- Configuring User Journeys to Ensure Correct Testing
- Executing Performance Tests
- Analyzing Performance Test Results
- Monitoring Underlying System Resources
- Customizing the Analysis Screens

If your installation is missing any user guides, please look at the “resources” section of the StressTester™ web site (www.stresstester.com) or contact Reflective Solutions Support (support@reflective.com).

# Introducing StressTester™

StressTester™ is an enterprise class performance testing tool utilized by companies of all sizes to test their mission critical applications.

It has been designed from first principles to be different from other tools; focusing on making the task of configuring performance tests as easy and fast as possible by removing the need to write test scripts.

The remainder of this section introduces StressTester™'s terminology, architecture and user interface.

## Terminology

| | |
|---|---|
| User Journey | A simulated business transaction.<br><br>This is often called a test script in other tools – but as StressTester™ is totally script-less, we use this term instead. |
| Step Group | A collection of steps (HTTP requests) that are grouped to form a unit of interest.<br><br>In most cases this is synonymous with a web page. |
| Dynamic Data | Variation of the data that is supplied as part of a request to the application. Consider a "Dynamic Data item" to be the equivalent of a variable or parameter. |
| Flow Control | Variation of the route through the User Journey's steps. This simulates different users taking different routes, looping differently, etc. – reproducing the variation that would happen in the real world. |

| | |
|---|---|
| Test Run Configuration | Details of a configuration of User Journeys, Injectors (the component that simulates user load) and Resource Monitors (which monitor the underlying system resources) to execute a performance test. |
| Resource Monitor | A measurement of a resource in the system under test. |

# Architecture

StressTester™ can be divided into four separate components.

### StressTester™ User Interface

The StressTester™ user interface is the control component, from which you can create, administer and review performance tests.

The capabilities available to you within the StressTester™ user interface depend on the license you possess. Where your license does not allow you to perform an action, the action will either not be accessible or you will be notified of your license limitation when you attempt the action.

### StressTester™ Database

StressTester™ uses a relational database as its repository – a major benefit of this is that the data is 'open'. Although we do not believe you will need to analyze it using any tool other than the StressTester™ user interface, if you do desire to do so, there are no additional modules required.

StressTester™ supports a number of database implementations – please refer to the Installation user guide for a complete list – and is by default installed with the Apache Derby database. This is the best choice for learning to use this application, but is not appropriate for large scale performance tests.

### Injectors

Injectors are the processes that simulate users accessing the application under test. They are designed to be highly scalable so many thousands of users can be simulated if required.

In addition, they are designed to require low system overheads (CPU, memory, etc.) so that you do not need to purchase special, excessively powerful hardware to simulate high loads.

### Monitor Agents

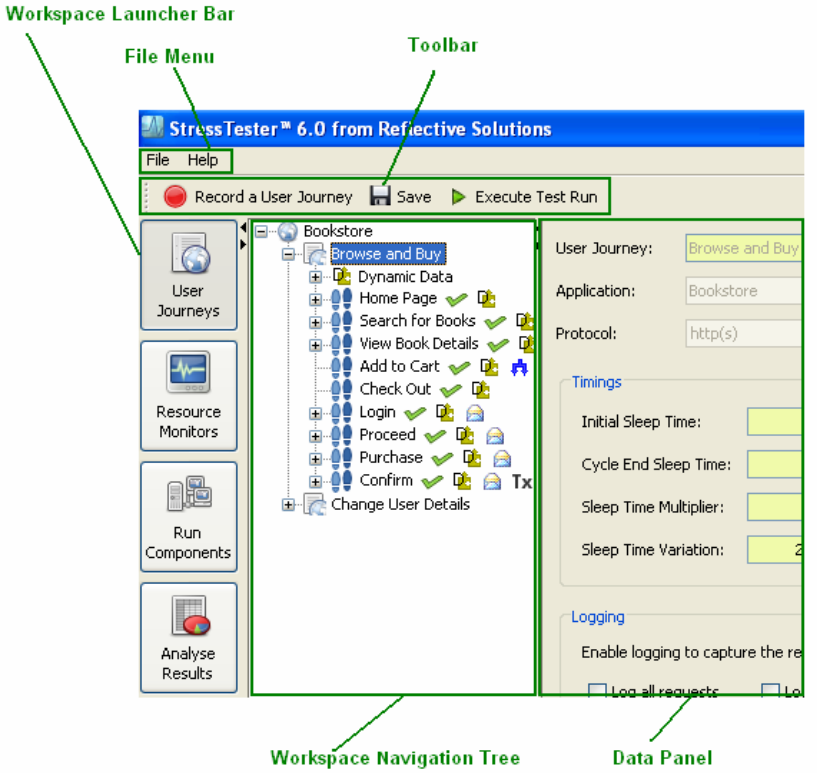
Monitor Agents are the processes that run Resource Monitors – which is turn measure resources within the system under test.

Resource Monitor results allow you to identify the cause for any identified poor performance and therefore where to focus effort in order to correct the problem.

The StressTester™ user interface is the only component you will need to use during this 30 minute introduction as it comes with a database and Injector embedded within it and this introduction does not cover monitoring system resources.

## User Interface

The diagram below shows the different areas of the StressTester™ user interface.



The StressTester™ user interface has four workspaces:

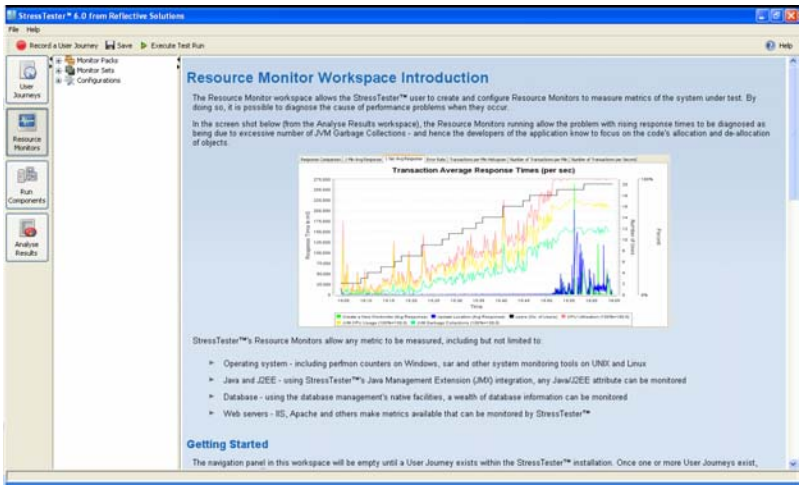
- User Journeys– record and configure User Journeys
- Resource Monitors – create and configure Resource Monitors
- Run Components – configure Injectors, and create and configure Test Run Configurations
- Analyse Results – analyze the results of performance tests and correlate with Resource Monitor results to understand the cause(s) of poor performance

You can switch between the workspaces by clicking the relevant button on the Workspace Launcher Bar.

## In-Tool Help

At various points within StressTester™ - typically when you are accessing a workspace or key feature for the first time – you will be presented with an information page.

These pages are easily identified as they have a blue background.




It is highly recommended that you read these pages when they are shown – they are intended to allow you to get up to speed with StressTester™ very quickly.

In addition, you can press the toolbar “Help” button at any time to see the context-sensitive help for the currently displayed Data Panel.

# Recording Your First User Journey

This section will show you how to record your first User Journey with StressTester™.

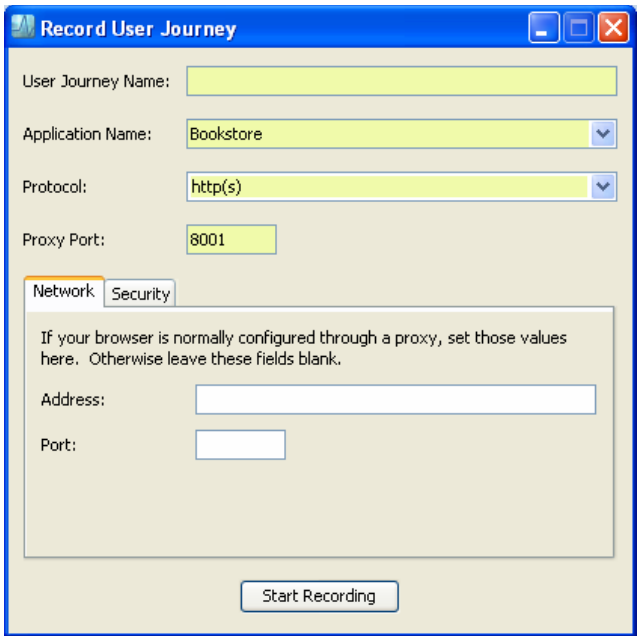
## Starting a Recording Session

To start a recording session, simply click the “Record a User Journey” button on the StressTester™ toolbar.

You will be presented with an in-tool information page that specifies that you need to consider whether to clear your browser’s cache, and also to configure your browser’s proxy to direct communications via StressTester™.

If you are unsure on how to perform either of these actions, please refer to the Recording HTTP User Journeys user guide.

When you close the information page you will be presented with the following pop-up screen.

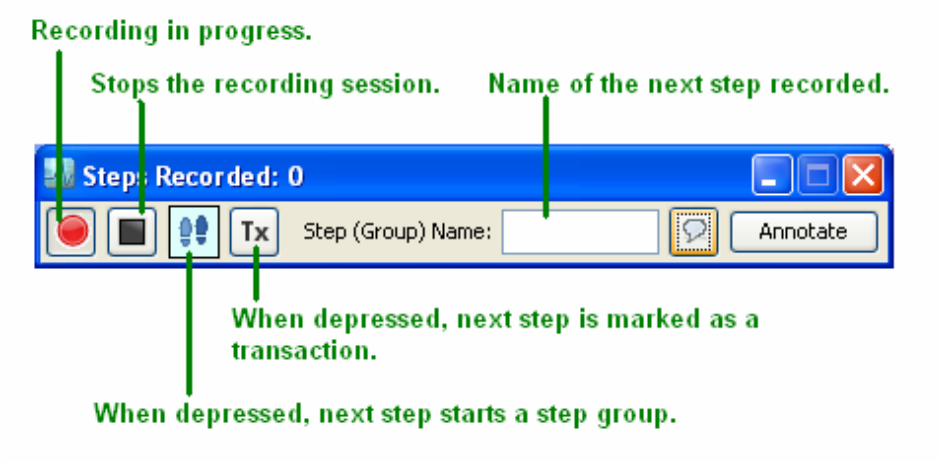



Simply enter a name for your User Journey, the name of the application you are recording against and, if your browser normally accesses the application using a proxy server, details of that server. Then press “Start Recording” to start the recording session.

# Annotation Window

StressTester™ will then display the annotation window – a small "on top" pop-up window that allows you to annotate the User Journey as you record it.

Although annotation during a recording is highly recommended, you do not need to worry if you forget to do so – all annotation actions can be performed after you have finished recording in the StressTester™ user interface.




# Recording Your User Journey

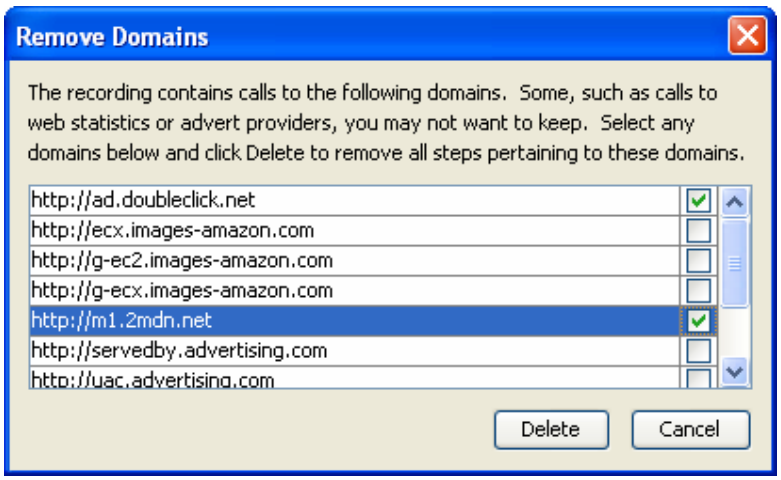
It is suggested you use the annotation window as follows:

1. Think about what you are about to do next in the browser
2. Type a name for the action in the "Step (Group) Name" field (e.g. "Open Home Page", "Search for Books", "Confirm Purchase")
3. If you wish this action to be counted as a transaction during performance tests, click the transaction button
4. Click "Annotate"
5. Perform the action in the browser

The above should be repeated for every action in the business transaction you record.

When you have finished recording your User Journey, click the stop button in the annotation window to end the recording session

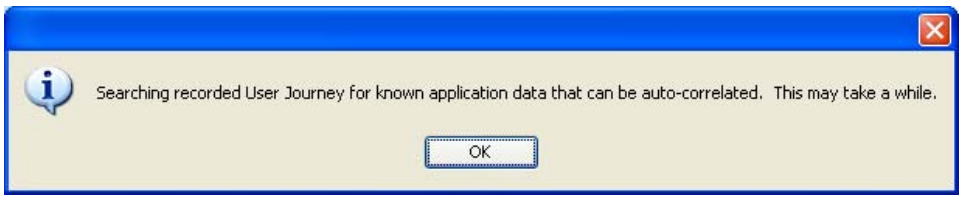
If your recorded User Journey accesses more than one domain, StressTester™ will present you with a post-recording pop-up window, listing the domains and allowing you to select any you do not wish to include in performance tests. Such domains are typically advert servers, site tracking systems, etc.



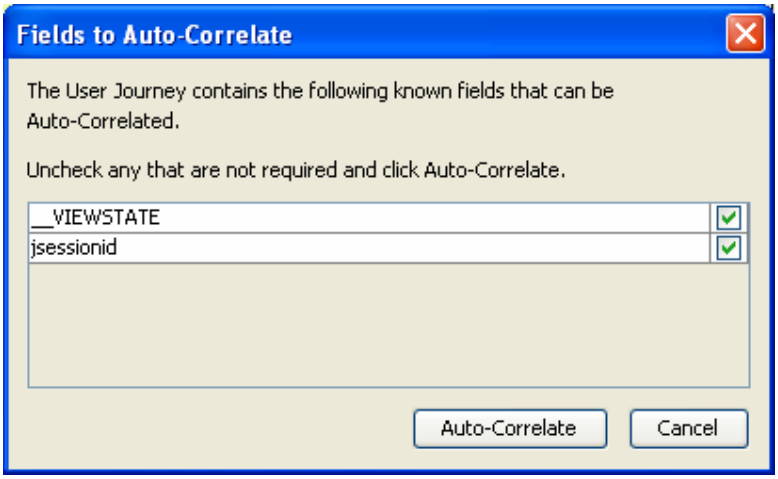

Select any domains you do not wish to include in performance tests and click the “Delete” button. This will delete any steps recorded that access the selected domains.

If you wish to keep all domains in your User Journey, just click the “Cancel” button.

StressTester™ will then present an informative pop-up indicating that it is about to search for known session control variables that it can handle automatically. These variables (such as __VIEWSTATE in .NET applications) are usually hidden from the user of an application but if they are not handled correctly, the test will likely fail.



Clicking “OK” will start the search. In the event that StressTester™ finds any known variables, it will present a pop-up window identifying those found.



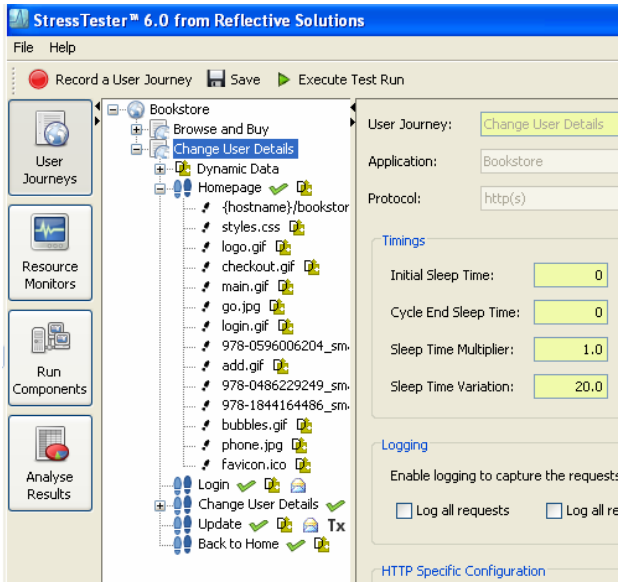

Select any variables that you wish StressTester™ to handle on your behalf and click the “Auto-Correlate” button. In the event that no known variables were found an informative pop-up will be presented.


Click the “OK” button to close.

**Note:** It is highly recommended, unless you are sure you do not need to do so, that you always allow StressTester™ to auto-correlate any session variables it finds.

# Inspecting Your User Journey

Following the recording session, StressTester™ will display the User Journeys workspace with the recorded User Journey selected.




In the example above, you can see that the recorded User Journey is called “Change User Details” and is used to test the “Bookstore” application.

The User Journey contains the pages (step groups): “Homepage”, “Login”, “Change User Details”, “Update” and “Back to Home”.

In addition, by clicking on the “Home Page” node, the list of requests (steps) that constitute the web page have been displayed.

Up to eight icons can appear after a page (or a step within a page) that provide a visual indication about the page or step’s details. In the example screen shown above, the ‘envelope’ icon shows that the “Login” and “Update” pages send POST data; in addition, the “Update” page is counted as a transaction when calculating transaction throughput figures.

Try clicking different nodes and look at the data panels displayed to investigate the User Journey you have recorded. For more information on the fields on the various data panels, or any other aspect of the User Journeys workspace, please refer to the Configuring User Journeys to Ensure Correct Testing user guide or the in-tool help.

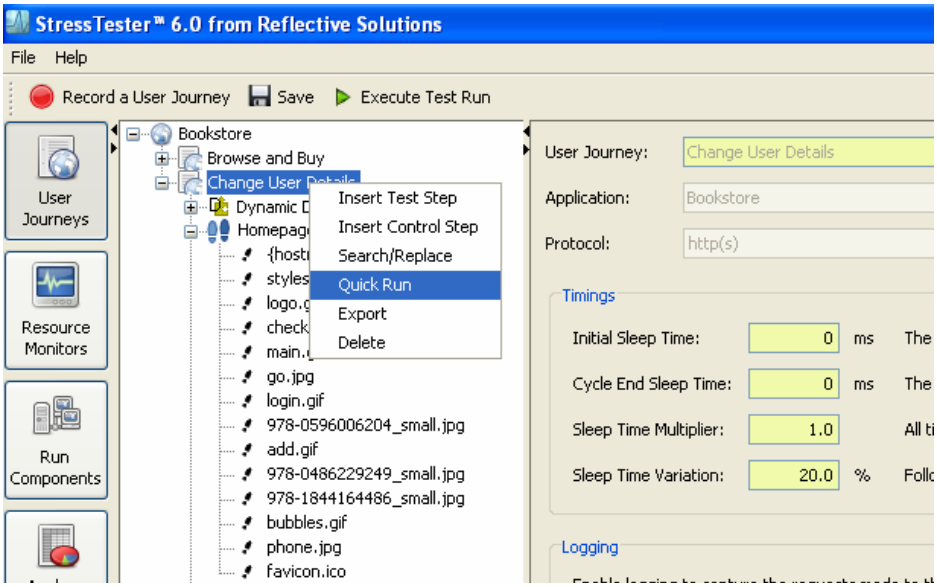
# Replaying Your User Journey

You are now ready to replay your User Journey to confirm it executes successfully.

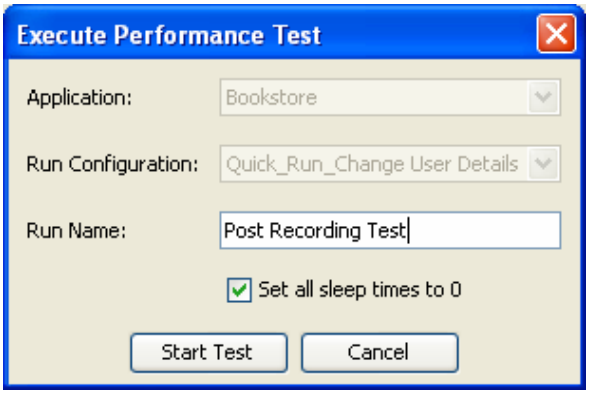
## Quick Run – Execute Your User Journey

Quick Run is the StressTester™ facility to easily execute a performance test that runs one User Journey, simulating one user, for one cycle (pass through the User Journey). It is therefore very useful when testing User Journeys during their configuration.

To Quick Run a User Journey, right-click on the User Journey node and select “Quick Run” as shown below.




You will then be presented with a pop-up window asking you for a name for the performance test that is about to be executed, as shown below.



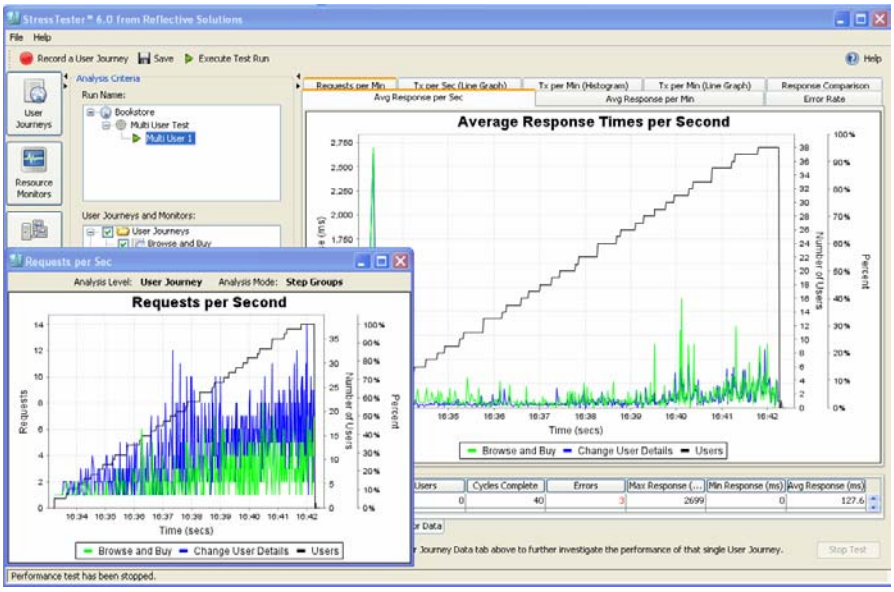

Enter a Run Name, and set the “Set all sleep times to 0” checkbox as required. When this control is checked, the recorded (inter-step) sleep times are ignored, and the User Journey executes as fast as possible. When the control is unchecked, the inter-step times are adhered to, and the User Journey executes at the speed at which it was recorded.

Click the “Start Test” button. At this point, StressTester™ will start the performance test and then switch you to the Analyse Results workspace and select the currently executing performance test.

You will see a screen similar to that shown below.

You should set the Refresh field to 5 seconds, and select the User Journey in the “User Journeys and Monitors” tree. This will cause data to be displayed within the graphs and tables on the screen, and automatically updated every 5 seconds.

You can display other graphs and data tables by selecting the appropriate tab, and can open many graphs and data tables at the same time by right-clicking a tab and selecting “Open in New Window”.




For further information on the analysis screens – including how to drill-down to individual page performance analysis and how to configure your own graphs and data tables – please refer to the Analyzing Performance Test Results user guide.

# Did It Run Successfully?

StressTester™ will display an information message after the quick run performance test has completed.

The first thing to check once the test has finished is whether any errors occurred. If you look at the bottom right area of the analysis screen, you will see at error count for the performance test – this should be 0.


If the error count is not zero, click the “Details” button to investigate the errors. Any reported errors should be addressed and if you do not know how to correct an error yourself, please contact Reflective

Solutions Support (support@reflective.com) who will be pleased to assist you.

The next thing to consider is whether the application responded as expected; it should be remembered that no checking of application responses (using the Validation Criteria field on the step properties screen) has as yet been specified.

Every quick run test logs each request made by StressTester™ and the application's response to a directory with the same name as the test within the `/log` directory of the StressTester™ installation.

If you sort the contents of this directory in ascending order, you will see one request file (file type of `request`) and one response file (file type of whatever the response type is) for each User Journey step.

For names steps, the name of the step appears within the file.

It is advised that you open each of the named response files and check that the contents of the files are that what you would expect to be returned from the application. Most of these files will be HTML and hence will open and display within a browser.

**Note:** Although the pages will display within a browser, files of type `html` will have no embedded images or other resources – these will be contained in subsequent step responses.

Once again, if you find any unexpected results, please investigate and feel free to contact Reflective Solutions Support for assistance.

# Configuring Validation Criteria

It is obvious that you would not want to check the result logs every time you executed a performance test – in fact for many thousands of simulated users, and many cycles test this would be impossible as the number of files would be very high.

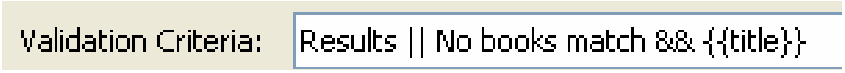
StressTester™ can be configured to perform runtime checks on the responses from the application under test using step Validation Criteria.

The suggested "rule of thumb" is that if it is worth naming a step in a User Journey, it is worth checking its response by specifying Validation Criteria.

Validation Criteria are configured on a step or Step Group's properties screen and can include:

- operators (AND is represented with `&&`, OR with `||` and NOT with `!`)
- complex expressions (all expressions are evaluated left to right)
- Dynamic Data items.

This allows Validation Criteria specifications to be very exact; ensuring the application is indeed returning what is expected.




In the example above, the step response must contain either the literal "`Results`" or "`No books match`" and the current value of the 'title' Dynamic Data item.

You can obviously use the logged responses from a Quick Run to choose the Validation Criteria.

It is advised that you now choose and specify Validation Criteria for each named step within your User Journey and execute the Quick Run performance test again to confirm these are correct.

# Simulating Multiple Users

The Quick Run used in the sections above is a very easy way to test a User Journey as it is being configured. However, the limitation is that Quick Run only simulates one user running one cycle of the User Journey.

Obviously, the purpose of performance testing is to simulate many users, performing different business transactions, and possibly from many different locations.

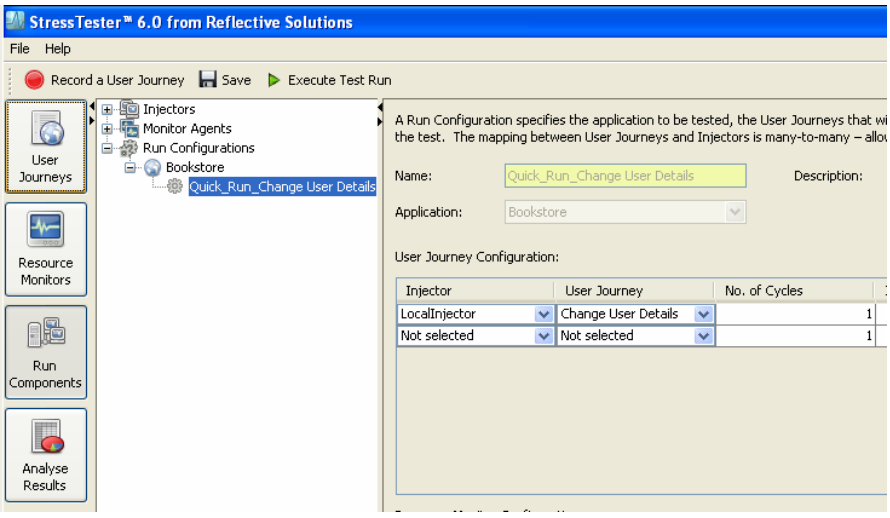
## Test Run Configurations

The description of how many simulated users will run each User Journeys, from what locations, and how the simulated user numbers change during the test, is configured within StressTester™ using a Test Run Configuration.

The Test Run Configuration screen is accessed by moving to the Run Components workspace.

If you explode the Run Configurations node in the navigation tree, you will see all the applications that have Test Run Configurations, and by exploding the application node you can see the configurations themselves.

Do this and you will see the Quick Run configuration for your User Journey – these are created automatically by StressTester™ the first time you Quick Run a User Journey.



What we will do next, is create a Test Run Configuration that ramps the simulated users on your User Journey (to a maximum of 3) over a period of a few minutes (30 seconds between each ramp), with each simulated user repeating the User Journey 5 times.

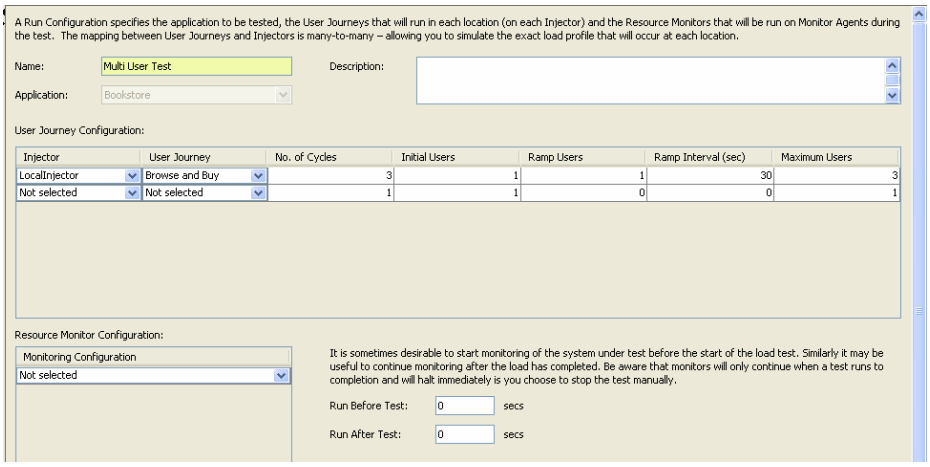
To create a new Test Run Configuration, right-click on the application node in the tree and select the “Create Run Configuration” option.

In the data panel displayed, give your configuration a name, and then choose the Injector (location) from which you wish to simulate load, and the User Journey you wish to run at that location.

**Note:** This is a true many-to-many relationship: an Injector can run more than one User Journey and a User Journey can run at more than one location (on more than one Injector).

Set the No. of Cycles to 5, the Initial Users to 1, Ramp Users to 1, Ramp Interval to 30, and Maximum Users to 3.

Your screen should look very similar to that shown below:




When you have completed the fields, click “Save” on the toolbar to save the Test Run Configuration.

# Executing a Multi-User Performance Test

It's time to test your new Test Run Configuration.

Right-click its node in the navigation tree and select the "Execute" option. In the pop-up window provide a name for the test and click "Start Test".

Once again you will be redirected to the Analyse Results workspace, with you currently running test selected.

Select the User Journey and click "Refresh", or set the Refresh field to a value greater than 0 for automatic refreshes.

You can now watch the performance test run, look at different graphs and data tables, and drill-down to single User Journey analysis if you wish (by double clicking the User Journey name in the data table).

# Now Look At …

We hope you have found this introduction guide of use and as always, would welcome any feedback you have – please simply email such to support@reflective.com.

Obviously, you have only just started seeing a small number of the features of StressTester™, and the User Journey you now have would not be suitable for performance testing as it does not yet correctly represent how real world users would perform the transaction (no variation of data, routes through the transactions, etc.).

It is recommended you now look at other provided user guides to further understand StressTester™'s features and benefits; and please remember – at any stage with any question – our Support people will be pleased to assist you.

## Correct Testing

In order to execute realistic performance tests, your User Journeys need to be configured to vary the data simulated users provide to the application under test, to vary the routes through the business transaction to simulate user indecision and looping, and to ensure that the inter-step sleep times are the same as they will be in the real world (including how they will vary in the real world).

StressTester™'s Dynamic Data and Flow Control facilities, in conjunction with the ease of setting and varying inter-step sleep times means it is an easy and quick job to make User Journeys correct.

You should refer to the Configuring User Journeys to Ensure Correct Testing user guide for information on all of these facilities.

## Analysis of Results

Obviously, the whole point of executing performance tests is to gain information about the application under test.

StressTester™'s powerful and configurable analysis screens allow any aspect of the test data to be represented in any graphical or

tabular format you require, in order that you can obtain the information your organisation requires.

You should refer to the Analyzing Performance Test Results and Customizing the Analysis Screens user guides for further details.

# Diagnosing Performance Issues

There are usually two possible outcomes from a performance test: the results show the application under test “passes” and can handle the load that was generated against it, or it “fails” and exhibits a performance problem.

In the latter outcome, sometimes the problem is obvious (a software component fails or starts to produce many errors), but often it is not so.

StressTester™’s Resource Monitors allow any aspect of the system under test to be monitored to provide the information that allows performance problems to be diagnosed.

Resource Monitors can measure metrics for: operating system, network, databases, J2EE or .NET platforms, web servers, load balancers and much more.

To understand how to configure and deploy Resource Monitors, refer to the Monitoring Underlying System Resources user guide.